

DAN LARIONESCU



Metode numerice



Jhe
10. Sept. 90

DAN LARIONESCU

METODE NUMERICE



EDITURA TEHNICĂ

București 1989



DAN LAFONTESCU

Lucrarea se înscrie în preocupările privind automatizarea calculului în activitatea de învățămînt, proiectare și cercetare, în promovarea mijloacelor electronice de calcul. În cele 22 paragrafe ale lucrării sînt succint trecute în revistă unele metode de calcul numeric ce permit studiul modelelor matematice pentru o mare diversitate de probleme ce pot fi întîlnite în practică. Un bogat material de figuri explicative, exemple numerice, subrutine scrise în limbajul FORTRAN, facilitează înțelegerea prezentării teoretice. Se acordă multă atenție studiului erorilor pe care le implică utilizarea metodelor numerice, se dau condiții de convergență a algoritmilor de calcul, se fac aprecieri privind stabilitatea numerică a acestora. Elementele de calcul numeric din prezenta lucrare sînt adresate și accesibile unui grup larg: cercurilor de informatică din licee, studenți ai institutelor tehnice, ingineri, economiști, programatori.

Redactor : Carmen Hotărănu
Tehnoredactor : Olimpiada Nistor
Coperta seriei : Constantin Guluță

Bun de tipar : 24.10.1989
Coli de tipar : 14
C.Z. : 517

ISBN 973-31-0106-0

L. P. „Gutenberg” Craiova, rd. 108

CUPRINS

Introducere	1
1. Algoritmi și erori de calcul	7
1.1. Algoritmi de calcul. Instabilitatea numerică a algoritmilor	7
1.2. Erori de calcul, propagarea erorilor	8
1.3. Scrierea subrutinelor de calcul în limbaj FORTRAN IV	12
2. Rezolvarea numerică a ecuațiilor algebrice și transeendente	15
2.1. Introducere	15
2.2. Metoda înjumătățirii intervalului	16
2.3. Metoda iterativă (metoda aproximațiilor succesive)	18
2.4. Metoda lui Newton (metoda tangențelor)	23
3. Rezolvarea sistemelor de ecuații liniare	27
3.1. Metoda lui Gauss	27
3.2. Metoda iterativă a lui Iacobi	32
3.3. Metoda Gauss-Seidel și metoda relaxării	35
4. Rezolvarea sistemelor de ecuații neliniare	37
4.1. Metoda iterativă a lui Iacobi	37
4.2. Metoda lui Newton	38
5. Metode numerice pentru calculul valorilor și vectorilor proprii	41
5.1. Introducere	41
5.2. Metoda puterii	41
5.3. Metoda Iacobi pentru matrici reale și simetrice	46
6. Criterii de aproximare a funcțiilor	50
6.1. Introducere	50
6.2. Criteriul de aproximare prin interpolare	52
6.3. Criteriul de aproximare cu abatere medie pătratică minimă	57
6.4. Criteriul de aproximare în sensul lui Cebîșev	59

CUPRINS

Introducere	1
1. Algoritmi și erori de calcul	7
1.1. Algoritmi de calcul. Instabilitatea numerică a algoritmilor	7
1.2. Erori de calcul, propagarea erorilor	8
1.3. Scrierea subrutinelor de calcul în limbaj FORTRAN IV	12
2. Rezolvarea numerică a ecuațiilor algebrice și transeendente	15
2.1. Introducere	15
2.2. Metoda înjumătățirii intervalului	16
2.3. Metoda iterativă (metoda aproximațiilor succesive)	18
2.4. Metoda lui Newton (metoda tangentelor)	23
3. Rezolvarea sistemelor de ecuații liniare	27
3.1. Metoda lui Gauss	27
3.2. Metoda iterativă a lui Iacobi	32
3.3. Metoda Gauss-Seidel și metoda relaxării	35
4. Rezolvarea sistemelor de ecuații neliniare	37
4.1. Metoda iterativă a lui Iacobi	37
4.2. Metoda lui Newton	38
5. Metode numerice pentru calculul valorilor și vectorilor proprii	41
5.1. Introducere	41
5.2. Metoda puterii	41
5.3. Metoda Iacobi pentru matrici reale și simetrice	46
6. Criterii de aproximare a funcțiilor	50
6.1. Introducere	50
6.2. Criteriul de aproximare prin interpolare	52
6.3. Criteriul de aproximare cu abatere medie pătratică minimă	57
6.4. Criteriul de aproximare în sensul lui Cebîșev	59

7. Aproximarea funcțiilor prin interpolare	61
7.1. Introducere	61
7.2. Polinomul de interpolare al lui Lagrange	62
7.3. Diferențe finite	68
7.4. Polinomul lui Newton de interpolare	71
7.5. Diferențe divizate. Polinoame de interpolare pe baza diferențelor divizate	77
8. Aproximarea funcțiilor cu abatere medie patratică minimă	81
8.1. Cazul continuu	81
8.2. Cazul discret	87
9. Aproximarea funcțiilor în sensul lui Cebîșev	98
9.1. Cazul continuu	98
9.2. Cazul discret	106
10. Sisteme liniare supradeterminate	109
10.1. Introducere	109
10.2. Soluție în sensul celor mai mici pătrate	109
10.3. Soluție în sensul lui Cebîșev (soluție de minimax)	112
11. Programare liniară	114
11.1. Introducere	114
11.2. Metoda simplex	118
12. Derivarea numerică	121
12.1. Introducere	121
12.2. Derivare numerică folosind polinoamele de aproximare	124
13. Integrarea numerică	128
13.1. Formule de cuadratură numerică	128
13.2. Metoda lui Romberg și tehnica de extrapolare a lui Richardson	133
13.3. Metoda dezvoltării în serie a funcției integrant	135
13.4. Metoda coeficienților nedeterminați	136
13.5. Integrarea numerică a integralelor improprii	138
13.6. Integrarea numerică a integralei nedefinite și a integralelor multiple	141
14. Integrarea numerică a ecuației diferențiale de ordinul întâi cu condiție inițială	144
14.1. Introducere	144
14.2. Metoda dezvoltării în serie Taylor	145

14.3. Metoda Runge-Kutta	147
14.4. Metode predictor-corrector	149
14.5. Metoda Picard	151
14.6. Studiul erorilor, metode relativ stabile	152
15. Integrarea numerică a sistemelor de ecuații diferențiale de ordinul întâi cu condiții inițiale	157
15.1. Introducere	157
15.2. Algoritme de calcul	158
16. Integrarea numerică a ecuațiilor diferențiale de ordin superior	161
16.1. Integrarea numerică a ecuațiilor diferențiale de ordin superior cu condiții inițiale	161
16.2. Integrarea numerică a ecuațiilor diferențiale de ordin superior cu condiții la limită	165
17. Rezolvarea numerică a ecuațiilor diferențiale cu derivate parțiale	169
17.1. Ecuații de tip eliptic	169
17.2. Ecuații de tip parabolic	173
17.3. Ecuații de tip hiperbolic	176
18. Ecuații recurente cu diferențe finite	177
18.1. Introducere	177
18.2. Ecuația omogenă de ordinul doi	178
18.3. Ecuația neomogenă de ordinul doi	180
19. Rezolvarea numerică a ecuațiilor integrale	182
19.1. Introducere	182
19.2. Integrarea ecuației tip Fredholm neomogenă de speța a doua prin metoda aproximațiilor succesive	184
19.3. Integrarea ecuației tip Volterra neomogenă de speța a doua prin metoda aproximațiilor succesive	185
20. Folosirea sumelor și seriilor în aproximări	187
20.1. Evaluarea aproximativă a sumelor seriilor numerice convergente	187
20.2. Transformarea seriilor numerice slab convergente în serii mai rapid convergente	190
20.3. Folosirea formulei lui Taylor și a seriei Taylor în aproximări	194

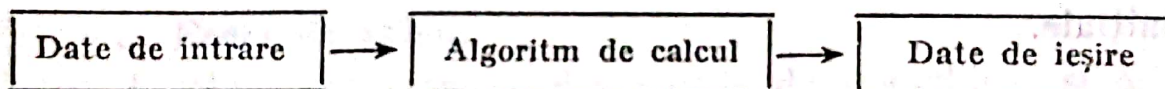
21. Metode probabilistice	199
21.1. Introducere	199
21.2. Procedee bazate pe valoarea medie	200
21.3. Procedee bazate pe simulare	200
22. Elemente ale limbajului de programare FORTRAN	202
22.1. Introducere	202
22.2. Principalele instrucțiuni ale limbajului FORTRAN	205
23. Exerciții propuse	216
Bibliografie	222

1. ALGORITMI ȘI ERORI DE CALCUL

1.1. Algoritmi de calcul.

Instabilitatea numerică a algoritmilor

Rezolvarea unei probleme impuse de practică presupune, pornind de la anumite date de intrare și folosind un algoritm de calcul, obținerea datelor de ieșire cerute, conform schemei logice :



Algoritmul de calcul reprezintă un sistem de reguli care transformă datele de intrare în date de ieșire cu ajutorul unor operații succesive, unic determinate.

Un algoritm trebuie să aibă următoarele proprietăți :

a) *Generalitate*. Se înțelege prin aceasta că algoritmul nu trebuie să rezolve numai o problemă, ci toate problemele din clasa respectivă.

b) *Finitudine*. Numărul de transformări intermediare, aplicate datelor de intrare pentru a obține datele de ieșire, este finit.

c) *Unicitatea*. Transformările intermediare trebuie să fie unic determinate.

Este importantă alegerea corectă a algoritmului de calcul. De exemplu, rezolvarea unui sistem de ecuații liniare

(în care numărul necunoscutelor este egal cu numărul ecuațiilor) se poate face folosind algoritmul dat de regula lui Cramer. Totuși, dacă sistemul are mai mult de 3 ecuații, evaluarea determinantilor implică în calcule reclamă un număr mare de operații aritmetice, astfel că, în acest caz, se recomandă să se folosească un alt algoritm de calcul.

$$\text{Fie sistemul : } 4,0000x + 0,8889y = 4,0000$$

$$1,0000x + 0,2222y = 1,0000$$

cu soluția unică $x = 1,0000$, $y = 0,0000$ obținută prin metoda eliminării. Dacă se reia sistemul de mai sus, lucrând însă doar cu trei cifre semnificative după virgulă pentru coeficienți, adică :

$$4,000x + 0,888y = 4,000 ; 1,000x + 0,222y = 1,000$$

același procedeu conduce la o infinitate de soluții. Se spune că algoritmul de calcul folosit este *instabil*.

Un algoritm de calcul este *stabil* dacă aplicat unei probleme cu date inițiale „ușor perturbate” conduce la o soluție apropiată într-un anumit sens, de soluția problemei cu date inițiale.

1.2. Erori de calcul, propagarea erorilor

Un algoritm de calcul este bine ales dacă precizia calculului este bună. Dar soluția depinde de o serie de erori dintre care amintim :

a) Erori inițiale

Erorile inițiale se formează din erori de măsurare datorate preciziei instrumentului (erori instrumentale) de obicei sistematice, dar și erori de observație care sînt neregulate sau întîmplătoare. Funcția de repartiție normală sau gaussiană care caracterizează erorile de observație are o formă caracteristică de clopot (fig. 1.1).

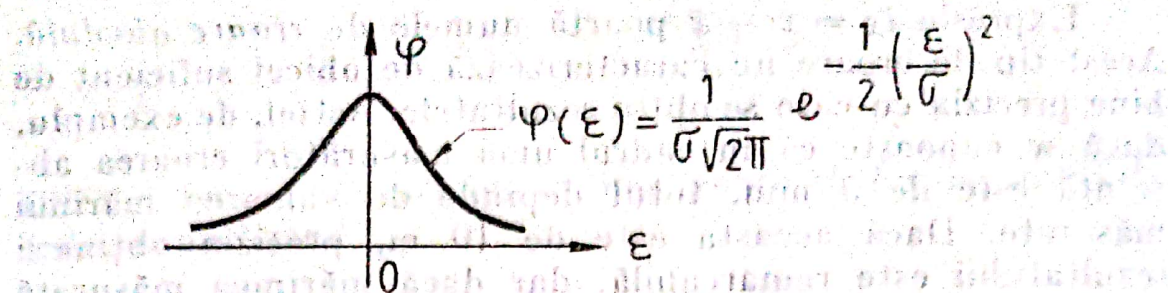


Fig. 1.1

Cu ajutorul acestei funcții, dacă se cunoaște dispersia σ pentru un anumit procedeu de măsurare, se poate calcula probabilitatea ca o eroare de observație să fie cuprinsă în intervalul $[a, b)$:

$$P(a \leq \varepsilon < b) = \int_a^b \varphi(\varepsilon) d\varepsilon$$

b) Erori de metodă

Pentru soluționarea numerică a unei probleme date se folosește o anumită metodă. Eroarea de metodă poate fi redusă printr-o alegere judicioasă a metodei de calcul.

c) Erori de calcul

Acestea sînt formate din erorile de trunchiere și erori de rotunjire.

Erorile de trunchiere apar de exemplu în aproximarea sumei unei serii printr-o sumă parțială, deci neglijînd ceilalți termeni care nu aparțin sumei parțiale.

Erorile de rotunjire apar datorită necesității de a folosi în calcule numere cu un număr finit de zecimale. În această situație numerele zecimale infinite se „rotunjesc”. Astfel, ultima zecimală păstrată rămîne neschimbată (rotunjire prin lipsă) dacă cifra care urmează după ea este cuprinsă între 0 sau 4. Ultima zecimală păstrată se majorează însă cu o unitate (rotunjire prin a laus) dacă după ea urmează 5, 6, 7, 8 sau 9.

Despre precizia rezultatelor :

fie x o valoare și $\bar{x}$ o valoare aproximativă a acesteia, rezultată în urma unei măsurători, a unei observații sau a unui calcul numeric.



Expresia $l_{\bar{x}} = x - \bar{x}$ poartă numele de *eroare absolută*. Acest tip de eroare nu caracterizează de obicei suficient de bine precizia cu care se obțin rezultatele. Astfel, de exemplu, dacă se cunoaște că în cadrul unei măsurători eroarea absolută este de 5 mm, totul depinde de valoarea mărimii măsurate. Dacă aceasta este de 10 m, precizia obținerii rezultatului este remarcabilă, dar dacă mărimea măsurată are valoarea de 1 cm, aceeași eroare absolută indică o precizie slabă a măsurării.

Precizia rezultatelor este mult mai bine pusă în evidență de eroarea relativă $r_{\bar{x}}$ care reprezintă raportul dintre eroarea absolută și valoarea aproximativă, adică $r_{\bar{x}} = l_{\bar{x}}/\bar{x}$. Rezultă : $x = \bar{x}(1 + r_{\bar{x}})$.

Se determină în cele ce urmează modul de propagare a erorilor absolute și relative în cele patru operații aritmetice.

Adunarea :

Se consideră valorile aproximative $\bar{x}$ și $\bar{y}$ ale valorilor adevărate x și y afectate de erorile $l_{\bar{x}}$ și $l_{\bar{y}}$. Avem :

$$l_{\bar{x}+\bar{y}} = (x + y) - (\bar{x} + \bar{y}) = (x - \bar{x}) + (y - \bar{y}) = l_{\bar{x}} + l_{\bar{y}}.$$

Rezultă :

$$l_{\bar{x}+\bar{y}} = l_{\bar{x}} + l_{\bar{y}}.$$

Scăderea :

$$l_{\bar{x}-\bar{y}} = (x - y) - (\bar{x} - \bar{y}) = (x - \bar{x}) - (y - \bar{y}) = l_{\bar{x}} - l_{\bar{y}}.$$

Deci :

$$l_{\bar{x}-\bar{y}} = l_{\bar{x}} - l_{\bar{y}}.$$

Necunoscînd semnul erorii absolute, nu putem trage concluzia că scăderea micșorează eroarea întotdeauna.

Înmulțirea :

$$l_{\bar{x}\bar{y}} = xy - \bar{x}\bar{y} = (\bar{x} + l_{\bar{x}})(\bar{y} + l_{\bar{y}}) - \bar{x}\bar{y} = \bar{x}l_{\bar{y}} + \bar{y}l_{\bar{x}} + l_{\bar{x}}l_{\bar{y}}$$

Neglijînd termenul $l_{\bar{x}}l_{\bar{y}}$ care este foarte mic, obținem :

$$l_{\bar{x}\bar{y}} \simeq \bar{x}l_{\bar{y}} + \bar{y}l_{\bar{x}}.$$

Împărțirea :

$$l_{\bar{x}/\bar{y}} = \frac{x}{y} - \frac{\bar{x}}{\bar{y}} = \frac{\bar{x} + l_{\bar{x}}}{\bar{y} + l_{\bar{y}}} - \frac{\bar{x}}{\bar{y}} = \frac{\bar{x} + l_{\bar{x}}}{\bar{y}} \cdot \frac{1}{1 + l_{\bar{y}}/\bar{y}} - \frac{\bar{x}}{\bar{y}}.$$

Ultimul factor din primul termen poate să fie dezvoltat în serie, considerind :

$$|l_{\bar{y}}/\bar{y}| < 1.$$

$$l_{\bar{x}/\bar{y}} = \frac{\bar{x} + l_{\bar{x}}}{\bar{y}} \left[1 - \frac{l_{\bar{y}}}{\bar{y}} + \left(\frac{l_{\bar{y}}}{\bar{y}} \right)^2 - \dots \right] - \frac{\bar{x}}{\bar{y}}.$$

Desfăcând paranteza și neglijând toți termenii ce conțin produse sau puteri mai mari decât 1 ale lui $l_{\bar{x}}$ și $l_{\bar{y}}$, obținem :

$$l_{\bar{x}/\bar{y}} \simeq \frac{1}{\bar{y}} l_{\bar{x}} - \frac{\bar{x}}{\bar{y}^2} l_{\bar{y}}.$$

Reluând calculele anterioare pentru erorile relative, se pot demonstra relațiile :

$$r_{\bar{x}+\bar{y}} \simeq r_{\bar{x}} \frac{\bar{x}}{\bar{x}+\bar{y}} + r_{\bar{y}} \frac{\bar{y}}{\bar{x}+\bar{y}}, \quad r_{\bar{x}\bar{y}} \simeq r_{\bar{x}} + r_{\bar{y}}$$

$$r_{\bar{x}-\bar{y}} \simeq r_{\bar{x}} \frac{\bar{x}}{\bar{x}-\bar{y}} - r_{\bar{y}} \frac{\bar{y}}{\bar{x}-\bar{y}}, \quad r_{\bar{y}/\bar{x}} \simeq r_{\bar{x}} - r_{\bar{y}}.$$

Se întâmplă adesea ca rezultatul unui șir de calcule să aibă o precizie redusă, deoarece pe parcursul efectuării acestor calcule s-a produs o scădere a doi termeni aproximativ egali. Într-adevăr, fie diferența : $52,287 - 51,939 = 0,348$. Să presupunem că pentru fiecare din termeni ne putem baza pe primele patru cifre, eroarea la ultima zecimală nedepășind o unitate : deci eroarea relativă a descăzutului și scăzătorului nu depășesc $1/50\,000$. Rezultatul scăderii, $0,348$ este nesigur la ultima zecimală cu două unități, fapt datorită căruia eroarea relativă poate fi de $2/348$, adică de 300 ori mai mare decât eroarea termenilor scăderii. Se recomandă în acest caz rescrierea algoritmului de calcul pentru evitarea acestor situații.

1.3. Scrierea subrutinelor de calcul în limbaj FORTRAN IV

În rezolvarea unei probleme impuse de practică, etapa principală este evident analiza logică și aducerea problemei la forma unui algoritm de calcul. În vederea utilizării unui calculator electronic etapa următoare impune rescrierea algoritmului de calcul într-un limbaj de programare, de exemplu FORTRAN IV, acceptat și folosit în prezent de fiecare centru de calcul. Se obține, în acest fel, scrisă pe formulare speciale de programare, lista de instrucțiuni ale programului principal. Algoritmii de calcul utilizați pentru rezolvarea diferitelor probleme conțin frecvent secvențe de calcul care se repetă des. Nu ar fi economic dacă aceste secvențe ar fi inserate în lista de instrucțiuni ale programului principal, de fiecare dată când apar. Este mai util în această situație să organizăm secvența respectivă de calcul ca un subprogram separat.

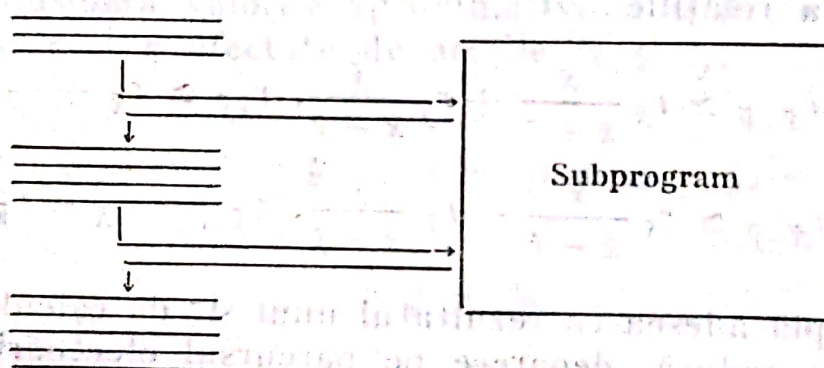


Fig. 1.2

În fig. 1.2, subprogramul a fost apelat și utilizat de programul principal reprezentat în partea stângă, de două ori. Folosirea subprogramelor determină o reducere a numărului de instrucțiuni ale programului principal. Limbajul de programare FORTRAN IV acceptă mai multe tipuri de subprograme, dintre acestea, complet fiind cel de tip SUBROUTINE care poate transmite programului principal mai multe valori calculate. Prima instrucțiune a unui subprogram tip SUBROUTINE conține în mod obligatoriu scris din poziția a șaptea a formularului de programare

cuvîntul de cod SUBROUTINE, urmat de numele subprogramului (maximum 6 caractere) și în continuare o listă de parametri fictivi. Urmează lista de instrucțiuni ale subprogramului, ultimele două fiind obligatorii RETURN și END, conform schemei, date în continuare.

5	6	7
		SUBROUTINE numele (lista de parametri fictivi)

		RETURN
		END

Instrucțiunile RETURN și END servesc pentru semnalarea poziției în care se face reîntoarcerea la programul principal, respectiv sfîrșitului subprogramului. Pentru apelarea unui subprogram tip SUBROUTINE, programul principal utilizează următoarea instrucțiune :

5	6	7

		CALL nume (listă de parametri efectivi)

Pentru exemplificarea modului în care programul principal folosește un subprogram tip SUBROUTINE, fie de exemplu subrutina numită TRANSP ce calculează transpusa unei matrici pătrate A, avînd n linii și n coloane.

Programul principal :

5	6	7

		CALL TRANSP (B, BT, 5)

Subprogram tip SUBROUTINE

5	6	7
1		SUBROUTINE <i>TRANSP</i> (<i>A</i> , <i>AT</i> , <i>N</i>) DIMENSION <i>A</i> (<i>N</i> , <i>N</i>), <i>AT</i> (<i>N</i> , <i>N</i>) DO 1 <i>I</i> = 1, <i>N</i> DO 1 <i>J</i> = 1, <i>N</i> <i>AT</i> (<i>I</i> , <i>J</i>) = <i>A</i> (<i>J</i> , <i>I</i>) RETURN END

Subrutina numită *TRANSP* folosește trei parametri fictivi, și anume *A*, *AT*, *N*, reprezentând, respectiv, matricea *A*, matricea A^T și numărul liniilor și coloanelor. În programul principal este definită matricea pătrată notată *B*, avind 5 linii și 5 coloane.

Atunci cînd în programul principal apare instrucțiune de apelare a subprogramului, **CALL** *TRANSP* (*B*, *BT*, 5), parametri efectivi *B*, *BT*, 5 iau locul parametrilor fictivi *A*, *AT*, *N*, se execută subprogramul, obținîndu-se în acest mod transpusa matricei *B*, după care la întîlnirea instrucțiunii **RETURN** această valoare este transmisă programului principal care din acest moment poate să o folosească. Un program complet pentru calculator presupune în mod obligatoriu un program principal însoțit, eventual, de o structură de subprograme. În capitolele următoare se scriu, drept aplicații, o serie de subrutine.

2. REZOLVAREA NUMERICĂ A ECUAȚILOR ALGEBRICE ȘI TRANSCENDENTE

2.1. Introducere

Fie $f: X \subset \mathbb{R} \rightarrow \mathbb{R}$ și ecuația $f(x) = 0$.

Dacă $f(x)$ este un polinom sau în urma unor transformări poate fi adusă la forma polinomială, ecuația se numește *algebrică*, în caz contrar o vom numi *transcendentă*. Rezolvarea ecuației $f(x) = 0$ implică parcurgerea a două etape.

a) *Separarea rădăcinilor ecuației necesitând determinarea unui interval în care se găsește fiecare din rădăcinile reale ale ecuației.*

Separarea rădăcinilor se poate face prin metoda șirului lui Rolle sau adeseori folosind datele problemei practice ce conduce la ecuația respectivă. Altă procedură constă în aproximarea funcției $f(x)$ printr-un polinom de interpolare (vezi capitolul 7), rădăcinile polinomului de interpolare servind ca valori aproximative inițiale pentru rădăcinile ecuației.

b) *Calculul aproximativ pentru fiecare rădăcină reală și evaluarea erorii.*

Să presupunem, de exemplu, că în intervalul (a, b) a fost separată o rădăcină reală δ pentru ecuația respectivă. Aceasta înseamnă că graficul funcției intersectează în δ axa Ox (fig. 2.1)

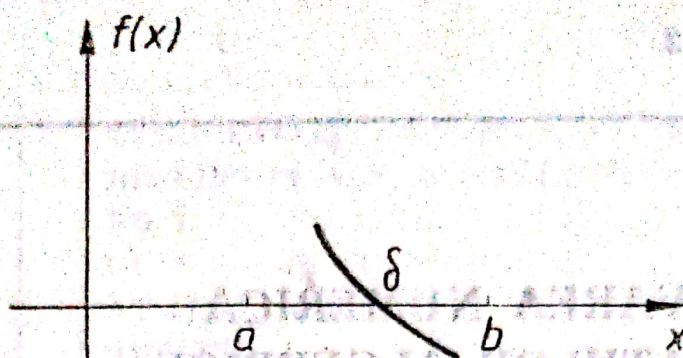


Fig. 2.1

Prin rădăcină *aproximativă* se înțelege o valoare $\tilde{x}$ suficient de apropiată de valoarea exactă δ . Cu alte cuvinte, dacă se dă o valoare acceptabilă ε a erorii, trebuie să avem :

$$|\tilde{x} - \delta| < \varepsilon, \varepsilon > 0.$$

Se prezintă în continuare câteva dintre metodele cele mai des utilizate pentru calculul aproximativ al rădăcinii.

2.2. Metoda înjumătățirii intervalului

Fie funcția continuă $f: [a, b] \rightarrow \mathbb{R}$ și ecuația $f(x) = 0$. În plus, se consideră că $f(a) \cdot f(b) \leq 0$.

În acest caz f are pe $[a, b]$ un număr impar de rădăcini (una cel puțin). Așa cum arată și numele, această metodă constă în înjumătățirea succesivă a intervalului $[a, b]$. Fie ε eroarea admisă.

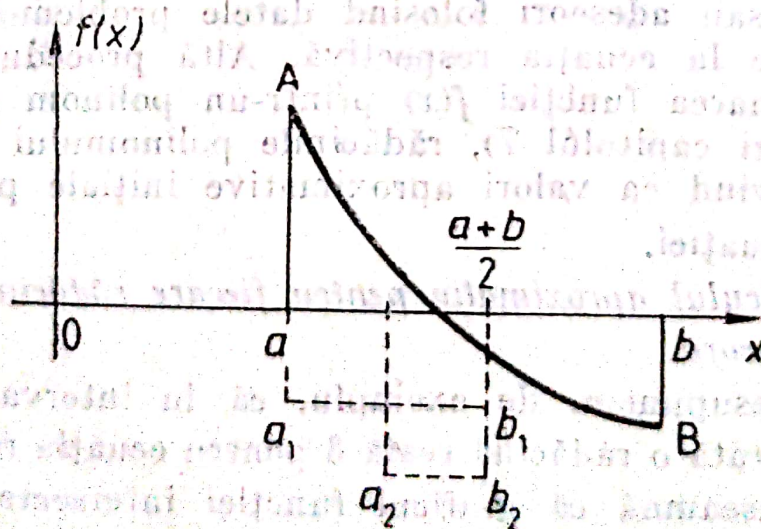


Fig. 2.2

Dacă $f\left(\frac{a+b}{2}\right) = 0$, atunci $\frac{a+b}{2} = \delta$ reprezintă chiar rădăcina căutată. Dacă nu, folosind punctul $\frac{a+b}{2}$, se împarte intervalul $[a, b]$ în jumătate (fig. 2.2). Se reține acea jumătate pentru care efectuând produsul valorilor funcției evaluate în capetele subintervalului, obținem o valoare mai mică decît zero. Aceasta este o indicație că rădăcina căutată se găsește în acel subinterval. Se notează cu $[a_1, b_1]$ subintervalul astfel determinat. Continuînd, se obține succesiunea de subintervale $[a_2, b_2], \dots, [a_n, b_n] \dots$ cu proprietățile: $f(a_n) \cdot f(b_n) < 0$, $b_n - a_n = \frac{1}{2^n}(b - a)$; $n = 1, 2, \dots$. Se obțin astfel șirurile (a_n) crescător sau constant și (b_n) șir descrescător sau constant. Avem $\delta = \lim_{n \rightarrow \infty} a_n = \lim_{n \rightarrow \infty} b_n$. După n înjumătățiri succesive, se poate alege ca valoare aproximativă pentru δ , valoarea:

$$\bar{x} = \frac{a_n + b_n}{2}.$$

Eroarea, care se comite dacă înlocuim pe δ prin $\bar{x}$, va fi:

$$|\varepsilon_{\bar{x}}| = |\delta - \bar{x}| \leq \frac{1}{2^n} |b - a|.$$

Metoda înjumătățirii intervalului este cea mai simplă dar și cea mai slab convergentă metodă pentru calculul aproximativ al rădăcinii unei ecuații date. Dacă au fost separate mai multe rădăcini, algoritmul expus anterior poate fi utilizat pentru calculul aproximativ al fiecărei rădăcini în parte. Dacă nu au fost separate rădăcinile în prealabil, algoritmul poate fi utilizat notînd pe

$$a_1 = -\infty, b_1 = 0 \quad \text{dacă} \quad f(-\infty) \cdot f(0) < 0$$

sau

$$a_1 = 0, b_1 = +\infty, \quad \text{dacă} \quad f(0) \cdot f(+\infty) < 0.$$

În practică simbolul ∞ va fi înlocuit cu o valoare mare, convenabil aleasă.

2.3. Metoda iterativă (metoda aproximațiilor succesive)

Se presupune că ecuația algebrică sau transcendentă $f(x) = 0$ are în intervalul $[a, b]$ o singură rădăcină reală δ . Se aduce ecuația la forma echivalentă $f(x) + x - x = 0$. Notînd $f(x) + x = \varphi(x)$, ecuația devine $x = \varphi(x)$. Se poate rescrie ecuația inițială în forma de mai sus, folosind și alte artificii de calcul. De remarcat că dacă δ este rădăcina ecuației inițiale $f(x) = 0$, atunci ea va fi rădăcină și pentru ecuația $x = \varphi(x)$. Fie $x_0 \in [a, b]$ o valoare aproximativă inițială pentru valoarea exactă δ . Metoda iterativă se definește printr-un șir de iterații succesive, astfel :

$$x_1 = \varphi(x_0), x_2 = \varphi(x_1), \dots, x_{k+1} = \varphi(x_k), \dots$$

Să presupunem că funcția $\varphi(x)$ este derivabilă pe intervalul $[a, b]$. Dacă

$$|\varphi'(x)| \leq \lambda < 1, (\forall)x \in [a, b],$$

atunci șirul de iterare definit mai sus converge către δ , rădăcină unică a ecuației $f(x) = 0$.

Într-adevăr, să considerăm diferența

$$x_{k+1} - \delta = \varphi(x_k) - \varphi(\delta).$$

Aplicînd teorema lui Lagrange, se obține:

$$x_{k+1} - \delta = \varphi'(\xi_k)(x_k - \delta); \xi_k \in (x_k, \delta).$$

Deoarece $|\varphi'(x)| \leq \lambda$, oricare ar fi $x \in [a, b]$,

$$|x_{k+1} - \delta| \leq \lambda |x_k - \delta|.$$

Dînd pe rînd lui n valorile 0, 1, 2, 3, ... se deduc succesiv :

$$|x_1 - \delta| \leq \lambda |x_0 - \delta|$$

$$|x_2 - \delta| \leq \lambda |x_1 - \delta| \leq \lambda^2 |x_0 - \delta|$$

$$|x_{k+1} - \delta| \leq \lambda^{k+1} |x_0 - \delta|.$$

Datorită acestei ultime inegalități și faptului că $0 \leq \lambda < 1$, rezultă :

$$\lim_{k \rightarrow \infty} x_{k+1} = \delta.$$

Deoarece $\varphi(x)$ este continuă, rezultă, prin trecere la limită în relația de iterare, $x_{k+1} = \varphi(x_k)$:

$$\delta = \lim_{k \rightarrow \infty} x_{k+1} = \varphi(\lim_{k \rightarrow \infty} x_k) = \varphi(\delta).$$

Se poate arăta că δ astfel obținută este unică. Într-adevăr, să presupunem că γ este o altă rădăcină a ecuației $f(x) = 0$ în intervalul $[a, b]$.

Atunci γ satisface egalitatea $\gamma = \varphi(\gamma)$. Să evaluăm diferența

$$\delta - \gamma = \varphi(\delta) - \varphi(\gamma) = (\delta - \gamma)\varphi'(\beta); \beta \in (\delta, \gamma)$$

$$\text{Deci } (\delta - \gamma)[1 - \varphi'(\beta)] = 0$$

Cum $1 - \varphi'(\beta) \neq 0$, deoarece s-a presupus $|\varphi'(x)| < 1$, $(\forall)x \in (a, b)$, rezultă că $\delta - \gamma = 0$, adică $\delta = \gamma$. Dacă $|\varphi'(x)| \leq \lambda < 1$, șirul de iterații succesive conduce spre soluția exactă δ , așa cum rezultă din următoarele două figuri, în funcție de forma curbei $y = \varphi(x)$.

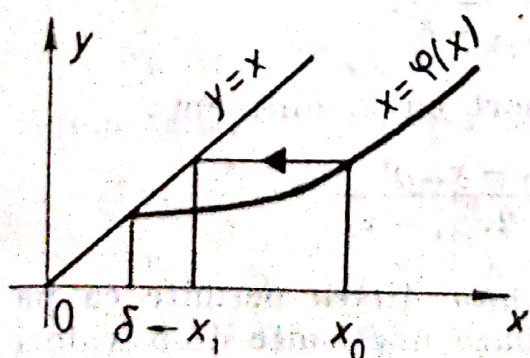


Fig. 2.3

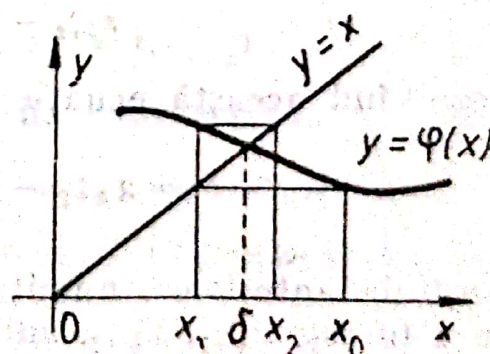
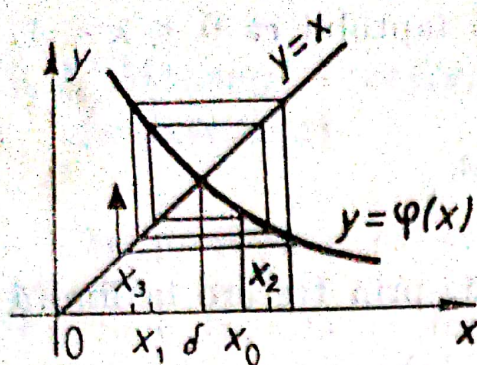


Fig. 2.4



În schimb, dacă $\lambda > 1$, șirul de iterații diverge, așa cum apare și în figura alăturată.

Pentru a urmări cum se propagă eroarea în calcule în procesul de iterare, să scriem dezvoltarea în serie Taylor a funcției $\varphi(x)$ în vecinătatea lui $x = \delta$:

$$\varphi(x) = \varphi(\delta) + \frac{x - \delta}{1!} \varphi'(\delta) + \dots$$

Să punem în această dezvoltare $x = x_k$ și apoi să o trunchiem după primii doi termeni:

$$\varphi(x_k) = \varphi(\delta) + \frac{x_k - \delta}{1!} \varphi'(\delta) \quad \text{sau :} \quad x_{k+1} = \delta + \frac{x_k - \delta}{1!} \varphi'(\delta).$$

Folosind notatiile $\varepsilon_k = x_k - \delta$ și $\varepsilon_{k+1} = x_{k+1} - \delta$, rezultă :

$$\varepsilon_{k+1} = \varepsilon_k \cdot \varphi'(\delta).$$

Ultima relație arată că eroarea în iterația $k+1$ este direct proporțională cu eroarea în iterația k . Se spune că gradul de convergență a metodei iterative este liniar. Folosind tehnica de extrapolare, rapiditatea convergenței metodei iterative se poate îmbunătăți astfel:

$$\varepsilon_{k+1} = \varphi'(\delta)\varepsilon_k \text{ sau } x_{k+1} - \delta = \varphi'(\delta)(x_k - \delta).$$

Avem, de asemenea, $x_{k+2} - \delta = \varphi'(\delta)(x_{k+1} - \delta)$.

Împărțind cele două relații de mai sus, obținem :

$$\frac{x_{k+1} - \delta}{x_{k+2} - \delta} = \frac{x_k - \delta}{x_{k+1} - \delta}.$$

Rezolvînd această ecuație în raport cu δ , obținem :

$$\delta = x_{k+2} - \frac{(x_{k+2} - x_{k+1})^2}{x_{k+2} - 2x_{k+1} + x_k}.$$

Formula anterioară numită și *relația Aitken* permite ca pe baza lui x_k, x_{k+1}, x_{k+2} , adică pe baza unei suite de 3 valori succesive ce aproximează pe δ , să se găsească o valoare îmbunătățită. Procedura de extrapolare poate fi folosită

pentru orice metodă numerică, cind se cunoaște modul de propagare a erorilor.

Metoda iterativă se poate utiliza cu succes și pentru studiul aplicațiilor numite *contracții*, într-un spațiu Banach (spațiu metric în care orice șir Cauchy este convergent).

Dacă X este un spațiu normat și A o parte a lui X , atunci aplicația $f: A \rightarrow A$ se numește *contracție* dacă există o constantă reală $0 \leq C < 1$ (numită *constantă de contracție*) astfel ca :

$$\|f(x) - f(y)\| \leq C \|x - y\|, (\forall)x, y \in A.$$

De exemplu, în $\mathbb{R}$ avem contracția dreptei reale definită astfel :

$$f(x) = Cx, 0 \leq C < 1.$$

Într-adevăr, folosind norma euclidiană, avem :

$$\begin{aligned} \|f(x) - f(y)\| &= |f(x) - f(y)| = |Cx - Cy| = \\ &= C|x - y|. \end{aligned}$$

Se numește *punct fix* al aplicației $f: A \rightarrow A$, orice soluție a ecuației $f(x) = x$, $x \in A$. Se poate arăta că orice contracție a unei submulțimi închise A dintr-un spațiu Banach are un punct fix unic, cu alte cuvinte, pentru orice contracție f a lui A , ecuația $f(x) = x$ admite o soluție unică. Într-adevăr, fie x_0 un punct oarecare al lui A . Folosind pe x_0 ca punct inițial, se formează șirul aproximațiilor succesive:

$$x_0, x_1 = f(x_0), x_2 = f(x_1) \dots x_n = f(x_{n-1}) \dots$$

Avem folosind definiția funcției de contracție :

$$\|x_1 - x_2\| = \|f(x_0) - f(x_1)\| \leq C \|x_0 - x_1\|$$

$$\|x_2 - x_3\| = \|f(x_1) - f(x_2)\| \leq C \|x_1 - x_2\| \leq$$

$$\leq C^2 \|x_0 - x_1\|$$

$$\|x_n - x_{n+1}\| \leq C^n \|x_0 - x_1\|$$

Oricare ar fi $n, p \in \mathbb{N}$, rezultă :

$$\begin{aligned} \|x_n - x_{n+p}\| &\leq \|x_n - x_{n+1}\| + \|x_{n+1} - x_{n+2}\| + \\ &+ \dots + \|x_{n+p-1} - x_{n+p}\| \leq \|x_0 - x_1\| (C^n + \\ &+ C^{n+1} + \dots + C^{n+p-1}) \leq \frac{C^n \|x_0 - x_1\|}{1 - C}. \end{aligned}$$

Deoarece $0 \leq C < 1$, pentru n suficient de mare, ultima valoare poate fi făcută mai mică decât orice $\varepsilon > 0$, ceea ce arată că şirul $(x_n)_{n \in \mathbb{N}}$ este un şir Conchy. Deoarece X este un spaţiu Banach rezultă că $(x_n)_{n \in \mathbb{N}}$ converge către o limită α , iar A fiind închisă, rezultă $\alpha \in A$. Deoarece f este contracţie, rezultă că este continuă, deci :

$$\alpha = \lim_{n \rightarrow \infty} x_n = \lim_{n \rightarrow \infty} f(x_{n-1}) = f(\lim_{n \rightarrow \infty} x_{n-1}) = f(\alpha).$$

În concluzie α este un punct fix al aplicaţiei f . Mai mult, se poate arăta că punctul α este unic. Într-adevăr, să presupunem prin absurd, că ar mai exista un punct fix β al contracţiei f . Atunci $\beta = f(\beta)$ şi deci :

$$\|\alpha - \beta\| = \|f(\alpha) - f(\beta)\| \leq C \|\alpha - \beta\|.$$

Rezultă : $\|\alpha - \beta\| (1 - C) \leq 0$.

Deoarece $\|\alpha - \beta\| > 0$, rezultă $1 - C < 0$, deci $C > 1$, ceea ce contrazice faptul că C este o constantă de contracţie. Se poate arăta uşor că o funcţie reală $f: [a, b] \rightarrow \mathbb{R}$ derivabilă pe $[a, b]$ şi avînd derivata $|f'(x)| \leq C < 1$, $(\forall)x \in [a, b]$ este un caz particular de contracţie. Într-adevăr, aplicînd teorema creşterilor finite, avem :

$$f(x) - f(y) = [f'(\xi)](x - y), \text{ cu } \xi \in (x, y),$$

deci $|f(x) - f(y)| \leq C |x - y|$.

În concluzie, cele demonstrate generalizează într-un spaţiu Banach problema determinării soluţiei unei ecuaţii de forma $x = f(x)$, folosind metoda aproximaţiilor succesive.

Aplicația 1

Se cere să se determine soluția reală a ecuației $x^3 - x^2 - x - 1 = 0$.

Șirul lui Rolle :

x	$-\infty$	$-1/3$	1	$+\infty$
	$-$	$-21/27$	-2	$+$

pune în evidență că ecuația admite o singură rădăcină reală în intervalul $(1, 2)$.

Se scrie ecuația sub forma echivalentă $x = 1 + 1/x + 1/x^2$, deci șirul iterațiilor este dat de $x_{k+1} = 1 + 1/x_k + 1/x_k^2$. Deoarece condiția de convergență este îndeplinită pentru intervalul $[1, 6]$, se poate porni șirul aproximațiilor succesive cu soluția de start 1, 6. Continuând șirul iterațiilor, găsim $x_1 = 2,015$, $x_2 = 1,740$, $x_3 = 1,904$, $x_4 = 1,800 \dots$ convergent către soluția exactă.

De remarcat că dacă am fi rearanjat ecuația inițială în felul următor: $x = x^3 - x^2 - 1$, și deci am fi folosit ca formulă de iterare expresia $x = x_k^3 - x_k^2 - 1$, atunci procedura diverge, condiția de convergență nefiind îndeplinită.

2.4. Metoda lui Newton (metoda tangentelor)

Fie ecuația algebrică sau transcendentă $f(x) = 0$ care are în intervalul $[a, b]$ o singură rădăcină reală δ . Să presupunem în plus că derivatele $f'(x)$ și $f''(x)$ sînt funcții continue și păstrează un semn constant pe intervalul $[a, b]$.

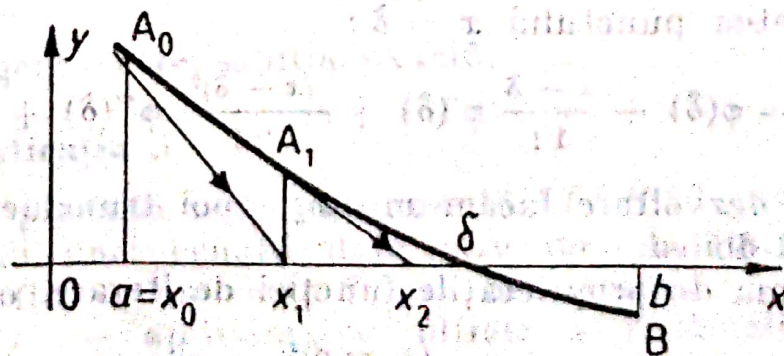


Fig. 2.6

Așa cum arată și numele, metoda tangentelor propune aproximarea rădăcinii δ prin șirul $x_1, x_2, \dots, x_n, \dots$ determinat prin intersecția tangentelor duse la curbă în punctele $A_0, A_1, \dots, A_n, \dots$ cu axa Ox (fig. 2.6). Fie punctul $A_k[x_k, f(x_k)]$. Ecuația tangentei la curba $y = f(x)$ în acest punct se scrie $y - f(x_k) = f'(x_k)(x - x_k)$. Să notăm prin x_{k+1} abscisa punctului de intersecție a acestei tangente cu axa Ox (deci $y = 0$). Se obține :

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}; \quad k = 0, 1, 2, \dots$$

Expresia de mai sus poartă numele de *formula iterativă a lui Newton*. Din figură se observă că valoarea aproximativă inițială x_0 a fost aleasă egală cu a , deci astfel ca $f(x_0) \cdot f''(x_0) > 0$. Aceasta este condiția alegerii punctului de start în metoda lui Newton, asigurându-se astfel ca $x_1 \in [a, b]$. Să notăm în continuare :

$$\varphi(x) = x - \frac{f(x)}{f'(x)}.$$

Funcția $\varphi(x)$ poartă numele de *funcție de iterare* a metodei lui Newton. Se verifică imediat că $\varphi(\delta) = \delta$, $\varphi'(\delta) = 0$, $\varphi''(\delta) \neq 0$.

Cu ajutorul funcției de iterare formula lui Newton devine :

$$x_{k+1} = \varphi(x_k), \quad k = 0, 1, 2, \dots$$

Se observă deci că metoda lui Newton este de fapt un caz particular al metodei aproximațiilor succesive.

Pentru a analiza cum se propagă eroarea în metoda lui Newton se dezvoltă funcția de iterare în serie Taylor în vecinătatea punctului $x = \delta$:

$$\varphi(x) = \varphi(\delta) + \frac{x - \delta}{1!} \varphi'(\delta) + \frac{(x - \delta)^2}{2!} \varphi''(\delta) + \dots$$

În această dezvoltare facem $x = x_k$, apoi trunchiem după termenul al doilea.

Ținând seama de proprietățile funcției de iterare, obținem :

$$\varphi(x_k) = \delta + \frac{(x_k - \delta)^2}{2!} \varphi''(\delta).$$

Deoarece $\varphi(x_k) = x_{k+1}$, putem scrie

$$x_{k+1} - \delta = \frac{\varphi''(\delta)}{2!} (x_k - \delta)^2$$

sau, notînd prin $(x_k - \delta)$, eroarea la pasul k , avem

$$\varepsilon_{k+1} = \frac{\varphi''(\delta)}{2!} \varepsilon_k^2$$

Această relație evidențiază că metoda lui Newton are gradul de convergență pătratic, adică eroarea la fiecare iterație este proporțională cu pătratul erorii de la iterația anterioară. Rezultă că metoda lui Newton converge mai repede decît metoda aproximațiilor succesive. Pe de altă parte, metoda lui Newton reclamă necesitatea evaluării funcției și a derivatei sale la fiecare iterație, fapt ce poate fi dificil sau chiar imposibil, dacă, de exemplu, funcția f nu este cunoscută analitic, ci prin intermediul unei tablele de valori. În practică alegerea între metoda aproximațiilor succesive și metoda lui Newton se face în funcție de forma lui f .

Aplicația 2

Să aplicăm metoda lui Newton la ecuația din aplicația 1 :

$$x^3 - x^2 - x - 1 = 0.$$

Formula de iterare a lui Newton este în acest caz

$$x_{k+1} = x_k - \frac{x_k^3 - x_k^2 - x_k - 1}{3x_k^2 - 2x_k - 1}.$$

Alegînd $x_0 = 1,6$ obținem :

$$x_1 = 1,905, \quad x_2 = 1,844 \dots$$

convergent către soluția exactă.

Aplicația 3

Se prezintă subrutina numită *ITERAT* pentru calculul rădăcinii unei ecuații date prin metoda iterativă. Parametrii fictivi au semnificația următoare :

XO — aproximația inițială a rădăcinii

- XF** — valoarea calculată a rădăcinii
N — numărul de iterații necesare pentru calculul rădăcinii cu eroarea admisă.
EPS — eroarea cu care se determină soluția.

5	6	7
	SUBROUTINE ITERAT (XO, XF, N, EPS)	
	DIMENSION X (N)	
	X (1) = XO	
20	DO 1 K = 2, N	
	X (K) = F (X(K-1))	
	IF (ABS (X (K) - X (K-1)) - EPS) 2, 2, 1	
1	CONTINUE	
	XF = X (N)	
	GO TO 3	
2	XF = X (K)	
3	WRITE (108,10) XF	
10	FORMAT (50X, 'X=', G 12.5)	
	RETURN	
	END	

În instrucțiunea 20, *F* reprezintă funcția de iterație pentru ecuația dată.

REZOLVAREA SISTEMELOR DE ECUAȚII LINIARE

3.1. Metoda lui Gauss

Metoda eliminării lui Gauss este una din cele mai vechi și mai des utilizate pentru rezolvarea sistemelor de ecuații liniare și pentru inversarea matricelor.

Se exemplifică această metodă pentru un sistem de 3 ecuații.

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3$$

Presupunem că matricea $A = (a_{ij})$ a sistemului este nesară (det $A \neq 0$). Atunci sistemul de mai sus are o soluție unică, care poate fi obținută prin algoritmul lui Gauss. Algoritmul lui Gauss implică o serie de pași succesivi de normalizare și reducere.

Pasul I (normalizare)

Înmulțim prima ecuație cu $1/a_{11}$. Celelalte ecuații rămân neschimbate. Notind noii coeficienți ai primei ecuații în urma acestei operații cu apostrof, sistemul de ecuații după primul pas devine:

$$x_1 + a'_{12}x_2 + a'_{13}x_3 = b'_1$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3$$

unde $a'_{1j} = a_{1j}/a_{11}$, $b'_1 = b_1/a_{11}$, $j = 2, 3$.

Pasul II (reducere)

Înmulțim prima ecuație cu $-a_{21}$ și o adunăm cu a doua. Înmulțim apoi prima ecuație cu $-a_{31}$ și o adunăm cu a treia. Notînd cu apostrof coeficienții astfel modificați, rezultă că sistemul capătă forma :

$$a_1 + a'_{12}x_2 + a'_{13}x_3 = b'_1,$$

$$a'_{22}x_2 + a'_{23}x_3 = b'_2,$$

$$a'_{32}x_2 + a'_{33}x_3 = b'_3,$$

unde $a'_{ij} = -a_{i1} + a_{ij}$, $b'_i = -a_{i1}b_1 + b_i$, ($i, j = 2, 3$).

Pasul III (normalizare)

Înmulțim ecuația a doua cu $1/a'_{22}$. Notînd cu două apostrofuri coeficienții astfel modificați, obținem :

$$x_1 + a'_{12}x_2 + a'_{13}x_3 = b'_1$$

$$x_2 + a''_{23}x_3 = b''_2$$

$$a'_{32}x_2 + a'_{33}x_3 = b'_3.$$

unde $a''_{2j} = a'_{2j}/a'_{22}$, ($j = 2, 3$) ; $b''_2 = b'_2/a'_{22}$.

Pasul IV (reducere)

Înmulțim ecuația a doua cu $-a'_{32}$ și o adunăm cu a treia. Notînd cu două apostrofuri coeficienții astfel modificați, sistemul devine:

$$x_1 + a'_{12}x_2 + a'_{13}x_3 = b'_1$$

$$x_2 + a''_{23}x_3 = b''_2$$

$$a''_{33}x_3 = b''_3$$

unde $a''_{33} = -a'_{32}a''_{23} + a'_{33}$, $b''_3 = -a'_{32}b''_2 + b'_3$.

Ca urmare a celor patru pași succesivi de normalizare și reducere, sistemul a fost adus la forma triunghiulară. În continuare se determină necunoscutele începînd cu ecuația 3 :

$$x_3 = b''_3/a''_{33}$$

$$x_2 = b''_2 - a''_{23}x_3$$

$$x_1 = b'_1 - a'_{12}x_2 - a'_{13}x_3$$

Se observă că în pașii de normalizare se efectuează împărțiri la anumiți coeficienți ai sistemului de ecuații. Dacă acești coeficienți numiți *pivoți* au valoarea zero sau chiar apropiată de zero, din motive de stabilitate numerică, trebuie să efectuăm rearanjarea ecuațiilor.

Se observă de asemenea că s-au folosit trei operații de bază fără să se modifice soluția sistemului. Acestea sunt :

- 1) înmulțirea ecuației cu o constantă ;
- 2) scăderea unei ecuații din alta și înlocuirea ecuației a doua cu rezultatul scăderii ;
- 3) rearanjarea ecuațiilor.

Metoda lui Gauss poate fi reluată lucrând numai asupra matricei extinse a sistemului :

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{pmatrix}$$

Acest fapt este util în special cînd se programează algoritmul lui Gauss pentru folosirea calculatoarelor electronice, deoarece limbajele specializate de programare au facilități speciale pentru utilizarea matricelor.

Operațiile din eliminarea gaussiană se realizează cu liniile din această matrice extinsă. Aceste operații sunt :

- 1) înmulțirea unei linii cu o constantă ;
- 2) scăderea unei linii din alta și înlocuirea celei de a doua cu rezultatul scăderii ;
- 3) rearanjarea liniilor.

Se observă corespondența dintre aceste operații și operațiile executate asupra ecuațiilor sistemului. În urma operațiilor impuse de cei patru pași de normalizare și reducere, matricea extinsă devine :

$$\begin{pmatrix} 1 & a'_{12} & a'_{13} & b'_1 \\ 0 & 1 & a''_{23} & b''_2 \\ 0 & 0 & a''_{33} & b''_3 \end{pmatrix}$$

Dacă se acționează în continuare asupra acestei matrice pentru a forma în primele 3 coloane matricea unitară (me-

toda Gauss-Jordan) atunci coloana a patra va reprezenta vectorul soluție :

$$\begin{pmatrix} 1 & 0 & 0 & x_1 \\ 0 & 1 & 0 & x_2 \\ 0 & 0 & 1 & x_3 \end{pmatrix}$$

Metoda eliminării lui Gauss poate fi utilizată și pentru calculul inversei unei matrice date. Astfel dacă se cere inversa matricei A

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

se formează matricea

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & 1 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 1 & 0 \\ a_{31} & a_{32} & a_{33} & 0 & 0 & 1 \end{pmatrix}$$

alăturind matricei A matricea unitară.

Algoritmul lui Gauss descris mai sus pentru a transforma primele trei coloane în matricea unitară va avea ca efect apariția, în următoarele trei coloane a matricei inverse lui A .

Aplicația 1

Se dă matricea nesingulară

$$A = \begin{pmatrix} 4 & 1 & -1 \\ 1 & 5 & -1 \\ 1 & 1 & 2 \end{pmatrix}$$

Se cere să se determine matricea inversă A^{-1} , folosind algoritmul lui Gauss.

Pe baza matricei A se formează matricea

$$\begin{pmatrix} 4 & 1 & -1 & 1 & 0 & 0 \\ 1 & 5 & -1 & 0 & 1 & 0 \\ 1 & 1 & 2 & 0 & 0 & 1 \end{pmatrix}$$

Aplicând algoritmul lui Gauss, se obține succesiv :

$$\text{Pasul 1 : } \begin{pmatrix} 1 & 0,25 & -0,25 & 0,25 & 0 & 0 \\ 1 & 5 & -1 & 0 & 1 & 0 \\ 1 & 1 & 2 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{Pasul 2 : } \begin{pmatrix} 1 & 0,25 & -0,25 & 0,25 & 0 & 0 \\ 0 & 4,75 & -0,75 & -0,25 & 1 & 0 \\ 0 & 0,75 & 2,25 & -0,25 & 0 & 1 \end{pmatrix}$$

$$\text{Pasul 3 : } \begin{pmatrix} 1 & 0,25 & -0,25 & 0,25 & 0 & 0 \\ 0 & 1 & -0,157 & -0,052 & 0,210 & 0 \\ 0 & 0,75 & 2,25 & -0,25 & 0 & 1 \end{pmatrix}$$

$$\text{Pasul 4 : } \begin{pmatrix} 1 & 0,25 & -0,25 & 0,25 & 0 & 0 \\ 0 & 1 & -0,157 & -0,052 & 0,210 & 0 \\ 0 & 0 & 2,367 & -0,211 & -0,157 & 1 \end{pmatrix}$$

$$\text{Pasul 5 : } \begin{pmatrix} 1 & 0,25 & -0,25 & 0,25 & 0 & 0 \\ 0 & 1 & -0,157 & -0,052 & 0,210 & 0 \\ 0 & 0 & 1 & -0,089 & -0,066 & 0,422 \end{pmatrix}$$

$$\text{Pasul 6 : } \begin{pmatrix} 1 & 0 & -0,210 & 0,263 & -0,052 & 0 \\ 0 & 1 & -0,157 & -0,052 & 0,210 & 0 \\ 0 & 0 & 1 & -0,089 & -0,066 & 0,422 \end{pmatrix}$$

$$\text{Pasul 7 : } \begin{pmatrix} 1 & 0 & -0,210 & 0,263 & -0,052 & 0 \\ 0 & 1 & 0 & -0,065 & 0,199 & 0,066 \\ 0 & 0 & 1 & -0,089 & -0,066 & 0,422 \end{pmatrix}$$

$$\text{Pasul 8 : } \begin{pmatrix} 1 & 0 & 0 & 0,244 & -0,065 & 0,088 \\ 0 & 1 & 0 & -0,065 & 0,199 & 0,066 \\ 0 & 0 & 1 & -0,089 & -0,066 & 0,422 \end{pmatrix}$$

$$\text{Rezultă : } A^{-1} = \begin{pmatrix} 0,244 & -0,065 & 0,088 \\ -0,065 & 0,199 & 0,066 \\ -0,089 & -0,066 & 0,422 \end{pmatrix}$$

3.2. Metoda iterativă a lui Iacobi

Fie un sistem de n ecuații liniare cu n necunoscute :

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2$$

$$a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n.$$

Metoda lui Iacobi începe prin a pune în evidență, în partea stângă a semnului egal, a variabilelor x_k . Cu alte cuvinte se rezolvă fiecare ecuație de rang k în raport cu x_k . Se obține :

$$x_1 = (b_1 - a_{12}x_2 - a_{13}x_3 - \dots - a_{1n}x_n)/a_{11}$$

$$x_2 = (b_2 - a_{21}x_1 - a_{23}x_3 - \dots - a_{2n}x_n)/a_{22}$$

$$x_n = (b_n - a_{n1}x_1 - a_{n2}x_2 - \dots - a_{n,n-1}x_{n-1})/a_{nn}$$

Pentru o asemenea rescriere a sistemului de ecuații trebuie ca toți coeficienții $a_{kk} \neq 0$. Dacă această condiție nu este îndeplinită, este necesară o schimbare a ordinii ecuațiilor sistemului dat, de asemenea o nouă notație a necunoscutelor, urmărind ca toate elementele diagonalei principale să fie diferite de zero. Pentru ușurința scrierii se trece la notația matricială :

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad \mathbf{u} = \begin{pmatrix} b_1/a_{11} \\ b_2/a_{22} \\ \vdots \\ b_n/a_{nn} \end{pmatrix}, \quad A = - \begin{pmatrix} 0 & a_{12}/a_{11} & \dots & a_{1n}/a_{11} \\ a_{21}/a_{22} & 0 & \dots & a_{2n}/a_{22} \\ \dots & \dots & \dots & \dots \\ a_{n1}/a_{nn} & a_{n2}/a_{nn} & \dots & 0 \end{pmatrix}$$

Sistemul devine : $\mathbf{x} = A\mathbf{x} + \mathbf{u}$.

Metoda lui Iacobi constă în faptul că pune sistemul de mai sus scris matricial într-o formă ce permite calculul iterativ : $x_{k+1} = Ax_k + u_k$ folosind un vector soluție inițială x_0 .

În practică este mult mai comod de folosit în locul relației matriciale de mai sus o relație referitoare la componente :

$$x_i^{(k+1)} = \frac{1}{a_{ii}} (b_i - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} x_j^{(k)}), \quad i = 1, 2, \dots, n.$$

Pentru stabilirea condițiilor în care algoritmul lui Jacobi converge, procedăm astfel : se folosește relația matricială iterativă pentru a obține :

$$x_1 = Ax_0 + u$$

$$x_2 = Ax_1 + u = A(Ax_0 + u) + u = A^2x_0 + Au + u = A^2x_0 + u(I + A)$$

$$x_k = A^k x_0 + u(I + A + A^2 + \dots + A^{k-1}).$$

Se poate demonstra că dacă toate valorile proprii ale matricei A sînt mai mici ca unitatea în modul, seria matricelor devine :

$$I + A + A^2 + \dots + A^k + \dots = (I - A)^{-1}.$$

Condiția de convergență a metodei lui Jacobi reclamă ca $\lim_{k \rightarrow \infty} A^k = 0$. Într-adevăr, dacă acest lucru se întîmpla tre-cînd la limită în relația ce definește algoritmul lui Jacobi, avem : $x = 0 + (I - A)^{-1}u$.

De unde $(I - A)x = u$, deci $x = Ax + u$, cu alte cuvinte $x = \lim_{k \rightarrow \infty} x_k$ reprezintă soluția sistemului. Dar condiția ca toate valorile proprii ale matricei A să fie mai mici decît unitatea în modul, implică condiția :

$$\sum_{\substack{j=1 \\ j \neq i}}^n \left| \frac{a_{ij}}{a_{ii}} \right| < 1, \quad 1 \leq i \leq n.$$

Relația de mai sus reprezintă pentru metoda iterativă a lui Jacobi o condiție necesară de convergență.

Dacă un sistem de ecuații liniare satisface condiția de convergență, deci dacă pentru fiecare ecuație coeficientul a_{ii}

de pe diagonala principală este în valoare absolută, mai mare decât suma valorilor absolute a celorlalți coeficienți, sistemul se numește *diagonal*. Multe dintre sistemele rezultate din probleme de fizică sînt de tipul diagonal. Sistemele diagonale se bucură deci de proprietatea că sînt rezolvabile prin metoda iterativă.

Aplicația 2

Se prezintă o subrutină de calcul după metoda Iacobi. Parametrii fictivi ai subrutinei sînt :

- N — numărul de ecuații
- A — matricea sistemului
- U — vectorul termenilor liberi
- X — vectorul necunoscutelor
- NIT — numărul de iterații

5	6	7
		SUBROUTINE IACOBI (N, A, U, X, NIT)
		DIMENSION A (N, N), U (N), X (N)
		DO 1 I = 1, N
1		Y (I) = X(I)
		DO 2 I = 1, NIT
		DO 3 J = 1, N
		V = 0
		DO 4 K = 1, N
		IF (J-K) 5, 4, 5
5		V = V + A (J, K) * Y (K)
4		CONTINUE
		X (J) = (U (J) - V) / A (J, J)
3		Y (J) = X (J)
7		FORMAT (50 X, G 18.8)
		WRITE (108,7) (X (J), J = 1, N)
2		CONTINUE
		RETURN
		END

3.3. Metoda Gauss-Seidel și metoda relaxării

Fie sistemul de ecuații :

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2$$

$$\text{-----}$$

$$a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n.$$

Procedind la fel ca la metoda Iacobi se rescrie acest sistem sub forma :

$$x_i = \frac{1}{a_{ii}} (b_i - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij}x_j), \quad i = 1, 2, \dots, n.$$

În contrast cu metoda Iacobi care nu folosește în iterații vectorul x_k pînă cînd toate cele n componente au fost determinate, metoda Gauss-Siedel folosește aceste componente pentru realizarea iterațiilor pe măsură ce ele sînt calculate, accelerînd astfel convergența procesului. Astfel, la pasul $(K + 1)$ se folosesc în calculul necunoscutei $x_i^{(k+1)}$ valorile necunoscutele $x_1^{(k+1)}, x_2^{(k+1)}, \dots, x_{i-1}^{(k+1)}$ deja calculate, deci :

$$x_i^{(k+1)} = \frac{1}{a_{ii}} (b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)}), \quad i = 1, 2, \dots, n.$$

Condiția de convergență pentru metoda lui Gauss-Seidel este aceeași ca și pentru metoda Iacobi. Se apreciază că la același nivel de precizie metoda Gauss-Seidel este de două ori mai rapidă. Metoda Gauss-Seidel poate fi modificată în continuare pentru sporirea vitezei de convergență a șirului iterațiilor. Pentru aceasta se scrie relația de mai sus astfel :

$$x_i^{(k+1)} = x_i^{(k)} + r_i^{(k)},$$

unde $r_i^{(k)}$, numit *reziduu*, reprezintă diferența dintre două iterații succesive calculate pentru x_i :

$$r_i^{(k)} = x_i^{(k+1)} - x_i^{(k)}.$$

Metoda iterativă definită prin algoritmul $x_i^{(k+1)} = x_i^{(k)} + \omega r_i^{(k)}$ este cunoscută cu numele de *suprarelaxare*. Parametrul de relaxare se alege astfel încît să crească viteza de convergență. Se poate arăta că $\omega \in (0, 2)$. Pentru $\omega = 1$ se regăsește metoda Gauss-Seidel. S-a găsit (în parte pe baza experiențelor făcute) că pentru o alegere potrivită a parametrului de relaxare, numărul aproximațiilor necesare a fi calculate poate fi redus în unele cazuri de o sută de ori.

Aplicația 3

În capitolul 17.1 rezolvînd problema lui Dirichlet pentru ecuația lui Laplace prin metoda rețelelor se ajunge la sistemul de ecuații :

$$x_1 = 1/4(0 + x_2 + x_3 + 0)$$

$$x_2 = 1/4(x_1 + 0 + 0 + x_4)$$

$$x_3 = 1/4(0 + x_1 + x_4 + 1)$$

$$x_4 = 1/4(x_2 + x_3 + 0 + 1)$$

Se cere să se rezolve sistemul de mai sus folosind metoda iterativă a lui Iacobi, de asemenea varianta Gauss-Seidel. Folosind un set de valori inițiale $x_1 = x_2 = x_3 = x_4 = 0$ metoda iterativă a lui Iacobi calculează iterativ următoarele seturi de valori ale necunoscutelor x_k .

Iterația	x_1	x_2	x_3	x_4
0	0	0	0	0
1	0	0	0,250	0,250
2	0,062	0,062	0,312	0,312
3	0,093	0,093	0,343	0,343

În varianta Gauss-Seidel imediat ce o necunoscută x_k este calculată, ea este utilizată în calculele următoare. Se ajunge la o tabelă de valori ce ilustrează convergența mai rapidă către soluția exactă a sistemului.

Iterația	x_1	x_2	x_3	x_4
0	0	0	0	0
1	0	0	0,250	0,250
2	0,062	0,078	0,328	0,351
3	0,101	0,113	0,363	0,369

REZOLVAREA SISTEMELOR DE ECUAȚII NELINIARE

4.1. Metoda iterativă a lui Iacobi

Fie un sistem de ecuații neliniare cu necunoscutele $x_1, x_2, \dots, x_n$;

$$\begin{aligned} f_1(x_1, x_2, \dots, x_n) &= 0 \\ f_2(x_1, x_2, \dots, x_n) &= 0 \\ f_n(x_1, x_2, \dots, x_n) &= 0 \end{aligned}$$

Folosind o notație vectorială, adică considerînd necunoscutele $x_1, x_2, \dots, x_n$ drept componente ale vectorului soluției x , sistemul poate fi rescris astfel: $f_i(x) = 0, 1 \leq i \leq n$.

Fie n funcții $F_i(x)$ derivabile parțial și alese astfel încît dacă $x_i = F_i(x)$, $1 \leq i \leq n$, aceasta implică $f_i(x) = 0, 1 \leq i \leq n$.

În particular, fie $\alpha_i = F_i(\alpha)$, $1 \leq i \leq n$, unde vectorul α reprezintă soluția sistemului de ecuații. Să presupunem că vectorul x_0 avînd componentele $x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}$ reprezintă o aproximație inițială a vectorului α . Metoda Iacobi calculează prin iterații succesive vectorul soluție folosind ca algoritm de calcul relațiile: $x_i^{(k+1)} = F_i(x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)})$, $i = 1, 2, \dots, n$ sau $x_i^{(k+1)} = F_i(x^{(k)})$, $i = 1, 2, \dots, n$.

Convergența acestei proceduri implică:

$$\sum_{j=1}^n \left| \frac{\partial F_i(x)}{\partial x_j} \right| \leq \lambda < 1.$$

În aceste condiții se poate demonstra că $\lim_{k \rightarrow \infty} x_k = \alpha$.
 Demonstrația se bazează pe folosirea formulei de medie.
 Avem :

$$\begin{aligned} x_i^{(k)} - \alpha_i &= F_i(x^{(k-1)}) - F_i(\alpha) = \\ &= \sum_{j=1}^n (x_j^{(k-1)} - \alpha_j) \frac{\partial F_i}{\partial x_j} [\alpha + \xi_i^{(k-1)}(x^{(k-1)} - \alpha)], \end{aligned}$$

unde $0 < \xi_i^{(k-1)} < 1$.

Ținând seama de condiția de convergență, obținem :

$$|x_i^{(k)} - \alpha_i| \leq h_{k-1} \sum_{j=1}^n \left| \frac{\partial F_i}{\partial x_j} \right| < \lambda h_{k-1} < h_{k-1},$$

unde s-a notat $h_{k-1} = \max_j |x_j^{(k-1)} - \alpha_j|$.

Prin inducție se arată că $|x_i^{(k)} - \alpha_i| \leq \lambda^k h_0$.

Trecînd la limită și ținînd seama că λ este subunitar, rezultă că într-adevăr $\lim_{k \rightarrow \infty} x_k = \alpha$, deci metoda Iacobi este convergentă.

4.2. Metoda lui Newton

Se ilustrează metoda pentru un sistem de două ecuații cu două necunoscute $f(x, y) = 0$, $g(x, y) = 0$ considerînd că funcțiile f și g sînt derivabile parțial într-o anumită vecinătate a punctului (x_0, y_0) . Valorile inițiale ale necunoscutelor sînt de obicei separate prin încercări sau prin mijloace grafice. Rezolvarea poate fi continuată apoi prin metoda tangentelor lui Newton.

Generalizînd metoda lui Newton unidimensională, se înlocuiesc suprafețele $f(x, y) = 0$ și $g(x, y) = 0$ prin planele lor tangente în vecinătatea punctului inițial și se ia ca aproximare următoare punctul în care dreapta de intersecție a celor două plane taie planul $z = 0$.



Ecuatiile planelor tangente în punctul (x_k, y_k) la suprafețele $z = f(x, y)$ și $z = g(x, y)$ sînt :

$$z = f(x_k, y_k) + (x - x_k)f'_x(x_k, y_k) + (y - y_k)f'_y(x_k, y_k)$$

$$z = g(x_k, y_k) + (x - x_k)g'_x(x_k, y_k) + (y - y_k)g'_y(x_k, y_k).$$

Să facem $z = 0$, apoi să notăm $x - x_k = h_k$, $y - y_k = r_k$.
Se obține în acest caz algoritmul iterativ al lui Newton:

$$x_{k+1} = x_k + h_k$$

$$y_{k+1} = y_k + r_k$$

unde h_k și r_k satisfac relațiile :

$$h_k f'_x(x_k, y_k) + r_k f'_y(x_k, y_k) = -f(x_k, y_k)$$

$$h_k g'_x(x_k, y_k) + r_k g'_y(x_k, y_k) = -g(x_k, y_k).$$

Extinderea la mai mult de două ecuații a procedurii de mai sus este imediată.

Aplicația 1

Se cere să se rezolve sistemul de ecuații $x = x^2 + y^2$, $y = x^2 - y^2$. Exceptînd soluția banală $x = 0$, $y = 0$, este evident, folosind metoda eliminării necunoscutele și șirul lui Rolle, că sistemul mai admite o soluție reală în vecinătatea punctului de coordonate $x = 0,8$ și $y = 0,4$.

Sistemul care determină pe h și pe r este următorul :

$$(2x_k - 1)h_k + 2y_k r_k = x_k - x_k^2 - y_k^2$$

$$2x_k h_k - (2y_k + 1)r_k = y_k - x_k^2 + y_k^2.$$

Luînd ca soluție inițială pe $x_0 = 0,8$ și $y_0 = 0,4$, pentru $k = 0$, sistemul de mai sus conduce la :

$$0,6h_0 + 0,8r_0 = 0$$

$$1,6h_0 - 1,8r_0 = -0,08.$$

Rezultă $h_0 = -0,027$, $r_0 = 0,020$.

Deci :

$$x_1 = x_0 + h_0 = 0,773$$

$$y_1 = y_0 + r_0 = 0,420,$$

Următoarele iterații vor da valori ale necunoscutelor mai apropiate de valoarea exactă.

Aplicația 2

Să se arate că sistemul de ecuații

$$x - 2y = 1$$

$$x - 1,999y = 0$$

este instabil.

Soluția sistemului de ecuații este $x = -1999$, $y = -1000$. Modificînd un singur coeficient al sistemului cu 2 miimi, acesta devine :

$$x - 2y = 1$$

$$x - 2,001y = 0,$$

avînd soluția $x = 2001$, $y = 1000$.

Se observă deci că mici modificări ale coeficienților sistemului au ca efect modificări foarte mari ale soluției acestuia.

Reprezentînd grafic fiecare ecuație a sistemului prin dreapta corespunzătoare, punctele de intersecție a acestor drepte (reprezentînd prin coordonatele lor soluția sistemului) - înt reprezentate în fig. 4.1.

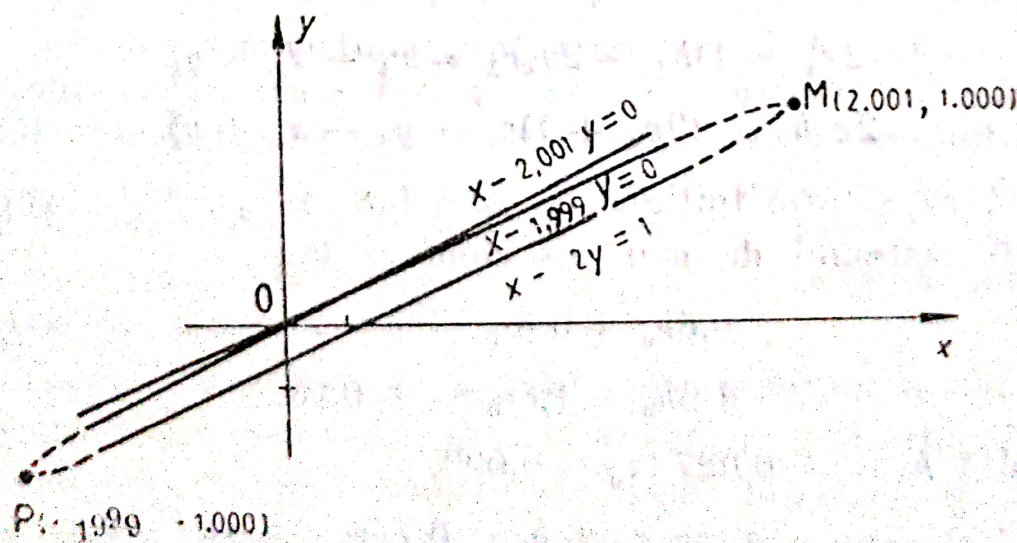


Fig. 4.1

METODE NUMERICE PENTRU CALCULUL VALORILOR ȘI VECTORILOR PROPRII

5.1. Introducere

Fie $A \in \mathbb{R}^{n \times n}$, o matrice reală cu n linii și n coloane. Un număr $\lambda \in \mathbb{C}$ se numește *valoarea proprie* a matricei A , dacă există un vector nenul $x \in \mathbb{C}^n$, astfel încât $Ax = \lambda x$. Vectorul x se numește *vector propriu* a lui A asociat valorii proprii λ . Ecuația de mai sus poate fi rescrisă sub forma $(A - \lambda I)x = 0$, unde I este matricea unitate. Ecuația matricială poate fi scrisă dezvoltat astfel:

$$\begin{pmatrix} a_{11} - \lambda & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} - \lambda & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} - \lambda \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Se obține un sistem de ecuații omogene care admite soluție nebanală numai dacă $\det(A - \lambda I) = 0$. Se observă că $\det(A - \lambda I)$ este un polinom de gradul n în λ , adică:

$$\det(A - \lambda I) = a_n \lambda^n + a_{n-1} \lambda^{n-1} + \dots + a_1 \lambda + a_0 = 0$$

Ecuația de mai sus este polinomială; polinomul din partea stângă a semnelui egal poartă numele de *polinom caracteristic*, iar ecuația în care intervine se numește *ecuația caracteristică* a matricei A .

Dacă privim sistemul de ecuații liniare $Ax = \lambda x$ ca pe o descriere a unui sistem fizic liniar a cărui cauză-efect este dată de matricea A , iar vectorul propriu x reprezintă datele de intrare în sistem, atunci forma sistemului arată că la ieșire se obține tot vectorul propriu x multiplicat prin constanta λ . Folosind vectorii proprii se poate „decupla” un sistem liniar, adică se poate realiza ca fiecare variabilă de ieșire să depindă numai de o variabilă de intrare. În teoria giroscopului se folosesc trei axe plasate pe direcțiile a trei vectori proprii independenți ai matricei momentelor de inerție în scopul obținerii unei forme simple a ecuațiilor dinamice.

Aplicația 1

Se cere să se calculeze valorile proprii și vectorii proprii pentru matricea :

$$A = \begin{pmatrix} 1 & 2 & 1 \\ 0 & -1 & -1 \\ 0 & 0 & 2 \end{pmatrix}$$

Pentru calculul valorilor proprii trebuie să rezolvăm ecuația

$$\begin{aligned} \det(A - \lambda I) &= \begin{vmatrix} 1 - \lambda & 2 & 1 \\ 0 & -1 - \lambda & -1 \\ 0 & 0 & 2 - \lambda \end{vmatrix} = \\ &= (1 - \lambda)(-1 - \lambda)(2 - \lambda) = 0 \end{aligned}$$

Obținem următoarele valori proprii distincte :

$$\lambda_1 = 2, \lambda_2 = 1, \lambda_3 = -1.$$

Introducând fiecare valoare proprie λ_k în sistemul $(A - \lambda I)x = 0$ obținem vectorul propriu x_k , corespunzător valorii proprii λ_k . Procedând astfel cu cele trei valori proprii găsite, obținem vectorii proprii: $x_1^T = \alpha(1, -1, 3)$, $x_2^T = \beta(1, 0, 0)$, $x_3^T = \gamma(1, -1, 0)$, $\alpha, \beta, \gamma \in \mathbf{R}$.

Sîntem deci în situația favorabilă de a avea 3 valori proprii distincte cărora le corespund 3 vectori proprii distincți. Nu aceasta este situația în general. Se poate arăta că orice matrice $A \in \mathbf{R}^{n \times n}$ are exact n valori proprii λ_k , $k =$

$= 1, 2, \dots, n$ în general complexe și nu neapărat distincte care coincid cu cele n rădăcini ale polinomului caracteristic. Valorile proprii complexe apar în perechi complex conjugate. De remarcat că matricea A din exemplul de mai sus este reală și simetrică. Se poate demonstra că orice matrice reală și simetrică având n linii și n coloane are toate valorile sale proprii reale, deși poate nu toate distincte.

Majoritatea problemelor de valori proprii care intervin în tehnică se referă la matrice reale și simetrice. Determinarea valorilor proprii pornind de la polinomul caracteristic nu este recomandabilă pentru matrice având mai mult de 3 linii și 3 coloane, efortul de calcul necesar pentru rezolvarea ecuației fiind exagerat de mare, Metodele practice de determinare a valorilor proprii sînt iterative.

5.2. Metoda puterii

Să presupunem că o matrice dată A de mărime $n \times n$, admite n valori proprii distincte $|\lambda_1| > |\lambda_2| > |\lambda_3| > \dots > |\lambda_n|$ care implică un sistem de n vectori proprii liniar independenți :

$$\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n.$$

Atunci un vector arbitrar $\mathbf{y}$ poate fi exprimat ca o combinație liniară cu vectorii proprii $\mathbf{y} = a_1\mathbf{x}_1 + a_2\mathbf{x}_2 + \dots + a_n\mathbf{x}_n$.

$$\begin{aligned} \text{Avem : } A\mathbf{y} &= a_1A\mathbf{x}_1 + a_2A\mathbf{x}_2 + \dots + a_nA\mathbf{x}_n = \\ &= a_1\lambda_1\mathbf{x}_1 + a_2\lambda_2\mathbf{x}_2 + \dots + a_n\lambda_n\mathbf{x}_n. \end{aligned}$$

Înmulțim în continuare cu A :

$$\begin{aligned} A^2\mathbf{y} &= a_1\lambda_1A\mathbf{x}_1 + a_2\lambda_2A\mathbf{x}_2 + \dots + a_n\lambda_nA\mathbf{x}_n = \\ &= a_1\lambda_1^2\mathbf{x}_1 + a_2\lambda_2^2\mathbf{x}_2 + \dots + a_n\lambda_n^2\mathbf{x}_n \end{aligned}$$

— — — —

$$\begin{aligned} A^p\mathbf{y} &= a_1\lambda_1^p\mathbf{x}_1 + a_2\lambda_2^p\mathbf{x}_2 + \dots + a_n\lambda_n^p\mathbf{x}_n = \\ &= \lambda_1^p \left[a_1\mathbf{x}_1 + a_2 \left(\frac{\lambda_2}{\lambda_1} \right)^p \mathbf{x}_2 + \dots + a_n \left(\frac{\lambda_n}{\lambda_1} \right)^p \mathbf{x}_n \right]. \end{aligned}$$

Deoarece valoarea λ_1 este dominantă, toate rapoartele tind spre zero pentru valori mari ale lui p . Avem deci:

$$A^p \mathbf{y}_1 \simeq \lambda_1^p a_1 \mathbf{x}_1$$

$$A^{p+1} \mathbf{y} \simeq \lambda_1^{p+1} a_1 \mathbf{x}_1.$$

Se știe că operația de împărțire a doi vectori nu are sens, dar se pot împărți componentele vectorilor, astfel că împărțind cele două relații de mai sus la nivel de componente, obținem valoarea proprie dominantă λ_1 . Mai mult, ultima valoare calculată $A^{p+1} \mathbf{y}$ va reprezenta, abstracție făcând de o constantă, o aproximare pentru vectorul propriu $\mathbf{x}_1$.

Aplicația 2

Se cere să se calculeze valorile proprii și vectorii proprii folosind metoda puterii pentru matricea:

$$A = \begin{pmatrix} 1 & 2 & 1 \\ 0 & -1 & -1 \\ 0 & 0 & 2 \end{pmatrix}$$

din aplicația 1.

Folosim ca vector de start pe $\mathbf{y} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$. Atunci

$$A\mathbf{y} = \begin{pmatrix} 4 \\ -2 \\ 2 \end{pmatrix}, A^2\mathbf{y} = \begin{pmatrix} 2 \\ 0 \\ 4 \end{pmatrix}, A^3\mathbf{y} = \begin{pmatrix} 6 \\ -4 \\ 8 \end{pmatrix}, A^4\mathbf{y} = \begin{pmatrix} 6 \\ -4 \\ 16 \end{pmatrix} \dots$$

$$A^9\mathbf{y} = \begin{pmatrix} 174 \\ -172 \\ 512 \end{pmatrix}, A^{10}\mathbf{y} = \begin{pmatrix} 342 \\ -340 \\ 1024 \end{pmatrix}.$$

Dacă împărțim componentele vectorului $A^{10}\mathbf{y}$ la componentele corespunzătoare ale vectorului $A^9\mathbf{y}$, obținem:

$$342/174 = 1,965$$

$$340/172 = 1,976$$

$$1024/512 = 2,000.$$

Rezultă că sintem destul de aproape de valoarea exactă $\lambda_1 = 2$.

Împărțind ultimul vector calculat prin 342, obținem: $\mathbf{x}_1^T \simeq 342(1 - 0,994 \ 2,994)$ o valoare apropiată de valoarea exactă $\mathbf{z} (1, -1,3)$ obținută prin rezolvarea ecuației caracteristice. Pentru a mări rapiditatea convergenței metodei se poate utiliza coeficientul Rayleigh. Coeficientul Rayleigh se definește ca $\mathbf{x}^T \mathbf{A} \mathbf{x} / \mathbf{x}^T \mathbf{x}$, unde notația $\mathbf{x}^T$ indică faptul că vectorul $\mathbf{x}$ a fost transpus. Dacă $\mathbf{A} \mathbf{x} = \lambda \mathbf{x}$, acest coeficient reprezintă exact valoarea proprie λ . Dacă $\mathbf{A} \mathbf{x} \simeq \lambda \mathbf{x}$, coeficientul Rayleigh va da doar o valoare aproximativă pentru λ . În anumite condiții șirul acestor coeficienți calculați pentru vectori generați prin metoda puterii, converge către λ_1 . Fie, de exemplu, $\mathbf{x}$ ultimul vector, calculat în secvența anterioară prin metoda puterii. Atunci:

$$\mathbf{A} \mathbf{x} = \begin{pmatrix} 686 \\ -684 \\ 2048 \end{pmatrix}, \quad \mathbf{x}^T \mathbf{A} \mathbf{x} = 2\,564\,324, \quad \mathbf{x}^T \mathbf{x} = 1\,281\,140 \text{ și co-}$$

eficientul Rayleigh este aproximativ 2,001 un rezultat corect pînă la a 3-a zecimală, deci convergența către λ_1 este mai rapidă decît efectuînd împărțirea componentelor a doi vectori calculați succesiv.

Algoritmul puterii poate fi utilizat și pentru a calcula valoarea proprie minimă λ_n . Într-adevăr, dacă matricea $\mathbf{A}$ este nesingulară, se știe că dacă $\mathbf{A} \mathbf{x} = \lambda \mathbf{x}$, atunci: $\mathbf{A}^{-1} \mathbf{x} = \lambda^{-1} \mathbf{x}$. Căutînd deci prin metoda puterii o valoare aproximativă pentru valoarea proprie maximă a lui $\mathbf{A}^{-1}$ se găsește de fapt o valoare aproximativă pentru inversa valorii proprii minime a lui $\mathbf{A}$. Metoda puterii poate fi adoptată pentru a găsi și valorile proprii intermediare între valoarea maximă și valoarea minimă. Într-adevăr se știe că dacă $\mathbf{A} \mathbf{x} = \lambda \mathbf{x}$, atunci $(\mathbf{A} - p\mathbf{I}) \mathbf{x} = (\lambda - p) \mathbf{x}$.

Aceasta înseamnă că $\lambda - p$ este o valoare proprie a matricei $\mathbf{A} - p\mathbf{I}$. De aici rezultă că printr-o alegere judicioasă a lui p (de exemplu $p = \lambda_1$ calculat) se poate determina orice valoare proprie a lui $\mathbf{A}$, prin algoritmul puterii. Alte valori

proprii pot fi obținute și prin modificarea vectorului de start. Fie vectorul arbitrar :

$$\mathbf{y} = a_1 \mathbf{x}_1 + a_2 \mathbf{x}_2 + \dots + a_n \mathbf{x}_n.$$

Dacă se alege vectorul de start astfel încît coeficientul a_1 să fie zero, în algoritmul puterii valoarea proprie λ_1 , respectiv vectorul propriu $\mathbf{x}_1$ nu vor mai apărea, rolul lor fiind preluat de λ_2 (următoarea valoare în ordinea mărimii), respectiv $\mathbf{x}_2$.

În privința numărului de iterații pe care le implică metoda puterii, calculele se opresc atunci cînd de la o iterație la alta diferența în lungimea vectorului propriu calculat este mică.

Astfel, dacă $l_{n-1} = \|A\mathbf{y}_{n-1}\|$ și $l_n = \|A\mathbf{y}_n\|$,

convergența metodei este presupusă dacă $\left| \frac{l_n - l_{n-1}}{l_{n-1}} \right| < \epsilon$,

unde ϵ este o valoare pozitivă dată, reprezentînd eroarea admisă.

5.3. Metoda Jacobi pentru matrice reale și simetrice

Dacă matricea P este nesingulară, atunci matricele A și PAP^{-1} au aceleași valori proprii, deoarece au aceeași ecuație caracteristică.

$$\begin{aligned} \text{Într-adevăr : } \det(PAP^{-1} - \lambda I) &= \det[P(A - \lambda I)P^{-1}] = \\ &= \det(P)\det(P^{-1})\det(A - \lambda I) = \det(A - \lambda I). \end{aligned}$$

În plus, dacă $\mathbf{x}$ este un vector propriu pentru matricea A atunci $P\mathbf{x}$ este un vector propriu pentru matricea PAP^{-1} .

Într-adevăr, deoarece $A\mathbf{x} = \lambda\mathbf{x}$, avem :

$$(PAP^{-1})P\mathbf{x} = PA\mathbf{x} = \lambda P\mathbf{x}.$$

Matricele A și PAP^{-1} se numesc *similare*, iar transformarea PAP^{-1} a lui A se numește *transformarea de similaritate*.

O rotație plană în planul (p, q) este definită prin matricea $R_{pq}(\varphi)$, $|\varphi| \leq \pi$ care este egală cu matricea unitate, cu ex-

cepția elementelor:

$$r_{pp} = r_{qq} = \cos \varphi, \quad r_{qp} = -r_{pq} = \sin \varphi.$$

De exemplu:

$$R_{12}(\varphi) = \begin{pmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Se observă că $R_{pq}^T(\varphi) = R_{pq}(-\varphi)$.

Dacă se face transformarea de similaritate

$$R_{pq}^T(\varphi) A R_{pq}(\varphi) = R_{pq}(-\varphi) A R_{pq}(\varphi)$$

atunci numai elementele situate pe liniile și coloanele p și q ale lui A vor fi afectate.

Metoda lui Iacobi pentru o matrice reală și simetrică A impune să se formeze un șir de matrice $A = A_0, A_1, A_2, \dots, A_k, \dots$, astfel ca matricea A_k să tindă către forma diagonală. În acest caz pe diagonala principală tind să apară valorile proprii ale matricei A . Pentru formarea lui A_k se folosesc transformări de similaritate pe bază de rotații plane. Astfel, dacă la pasul k , cel mai mare element (în modul) nedijagonal al matricei A_k este $a_{pq}^{(k)}$, atunci se efectuează transformarea:

$$A_{k+1} = R_{pq}(-\varphi) A_k R_{pq}(\varphi),$$

unde unghiul φ se alege astfel încât elementul $a_{pq}^{(k+1)}$ să fie egal cu zero. Se verifică ușor ca:

$$a_{pq}^{(k+1)} = a_{qp}^{(k+1)} = (a_{qq}^{(k)} - a_{pp}^{(k)}) \sin \varphi \cos \varphi + a_{qp}^{(k)} (\cos^2 \varphi - \sin^2 \varphi)$$

Alegînd unghiul φ astfel ca: $\operatorname{tg} 2\varphi = \frac{2a_{pq}^{(k)}}{a_{pp}^{(k)} - a_{qq}^{(k)}}$ rezultă $a_{pq}^{(k+1)} = a_{qp}^{(k+1)} = 0$. Fiecare pas al algoritmului lui Iacobi produce prin urmare o pereche de elemente nedijagonale egale cu zero. Din nefericire pasul următor, cînd produce o nouă pereche de zerouri, introduce valori diferite de zerouri în pozițiile precedente. Cu toate acestea șirul matricelor $\{A_k\}$ produce termeni din ce în ce mai apropiați de matricea diagonală.

Pentru caracterizarea apropierei matricei A de o matrice diagonală, se folosește suma σ a patratelor elementelor diagonale ale lui A . Suma patratelor elementelor diagonale ale lui A sînt calculate înainte și după fiecare iterație k . Fie aceste valori σ_{k-1} , respectiv σ_k . Criteriul pentru terminarea calculelor prin procedura Iacobi are loc dacă pentru un ε dat, avem: $1 - \frac{\sigma_{k-1}}{\sigma_k} < \varepsilon$.

Dacă se dorește și vectorii proprii, notînd prin U_k matricea de rotație folosită la pasul k , se formează matricea: $X_k = U_1 U_2 \dots U_k$ care va tinde către matricea vectorilor proprii.

Aplicația 3

Se cere determinarea valorilor și vectorilor proprii prin metoda lui Iacobi pentru matricea

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

reală și simetrică.

Alegînd la pasul 1 drept cel mai mare element nediagonal pe $a_{12} = 1$, deci $p = 1$, $q = 2$ și avem:

$$\operatorname{tg} 2\varphi = \frac{2a_{12}}{a_{11} - a_{22}} = \frac{2}{0}$$

Rezultă $2\varphi = \pi/2$, adică $\varphi = \pi/4$.

Deci $\sin \varphi = \cos \varphi = 1/\sqrt{2}$

Matricea A_1 este de forma

$$\begin{aligned} A_1 &= R_{12}(-\varphi) A R_{12}(\varphi) = \\ &= \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 2 & 0 & \sqrt{2} \\ 0 & 0 & 0 \\ \sqrt{2} & 0 & 1 \end{pmatrix} \end{aligned}$$

Pentru pasul al doilea alegem elementul nediagonal $a_{13} = \sqrt{2}$.
 Rezultă $\operatorname{tg} 2\varphi = 2\sqrt{2}$, $\sin \varphi = 0,577$, $\cos \varphi = 0,816$.
 Deci :

$$A_2 = R_{13}(-\varphi)A_1R_{13}(\varphi) =$$

$$= \begin{pmatrix} 0,816 & 0 & 0,577 \\ 0 & 1 & 0 \\ -0,577 & 0 & 0,816 \end{pmatrix} \cdot A_1 \cdot \begin{pmatrix} 0,816 & 0 & -0,577 \\ 0 & 1 & 0 \\ 0,577 & 0 & 0,816 \end{pmatrix}$$

Se obține astfel un șir de matrice A_k convergent către matricea A .

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 3 \end{pmatrix}$$

avînd pe diagonala principală valorile proprii ale matricei A .

CRITERII DE APROXIMARE A FUNCȚIILOR

6.1. Introducere

În foarte multe aplicații practice apare necesitatea aproximării unei funcții $f: [a, b] \rightarrow \mathbf{R}$ printr-o altă funcție $F(x)$ relativ simplă, astfel ca pentru orice valoare a lui x , valoarea lui $F(x)$ să fie „suficient de aproape” de valoarea lui $f(x)$.

Vom scrie $f(x) \simeq F(x)$, $x \in [a, b]$.

Există în special două cazuri în care se impune aproximarea funcției $f(x)$. Primul este acela în care funcția $f(x)$ are o expresie complicată sau este dificil de evaluat sau de manipulat în calcule. Astfel, de exemplu, pentru evaluarea funcției $\cos x$ prin operații aritmetice se impune mai întâi aproximarea funcției printr-o sumă parțială a seriei de puteri:

$$\cos x \simeq 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots (-1)^n \frac{x^{2n}}{(2n)!}$$

Al doilea caz în care se impune aproximarea funcției $f(x)$ este acela în care aceasta este dată printr-o tabelă de valori, obținută, de exemplu, ca urmare a unor măsurători:

$$\begin{array}{c|c} x_k & x_0 \quad x_1 \dots x_n \\ \hline f(x_k) & f(x_0) f(x_1) \dots f(x_n) \end{array}$$

În această situație se aproximează funcția dată tabelar printr-o expresie analitică, care să permită interpolarea în

tabela de valori, cu alte cuvinte estimarea valorilor $f(x)$ pentru $x \neq x_k$.

Fie $M = \{f|f: [a, b] \rightarrow \mathbb{R}\}$ un spațiu vectorial și fie o mulțime de funcții $\varphi_0(x), \varphi_1(x), \dots, \varphi_k(x)$ aparținând lui M , liniar independente, adică dacă $c_0\varphi_0(x) + c_1\varphi_1(x) + \dots + c_k\varphi_k(x) = 0$, să rezulte $c_0 = c_1 = \dots = c_k = 0$.

Aproximarea unei funcții f oarecare din M se face printr-o combinație liniară de un număr finit m de funcții de tipul φ_k , adică $f(x) \simeq F_m(x)$, unde $F_m(x) = c_0\varphi_0(x) + c_1\varphi_1(x) + \dots + c_m\varphi_m(x) = \sum_{k=1}^m c_k\varphi_k(x)$. Vom numi funcția $F_m(x)$ polinom

generalizat, aproximarea funcției f făcându-se în acest caz prin polinoame generalizate.

Foarte frecvent în procesul de aproximare se iau drept set de funcții liniar independente, funcțiile $1, x, x^2, \dots, x^m$. În acest caz, polinomul de aproximare $F_m(x)$ va fi un polinom algebric. Polinoamele sînt ușor de evaluat, iar suma, diferența și produsul a două polinoame conduc de asemenea la polinoame. În plus, polinoamele pot fi derivate și integrate cu ușurință. Aproximarea polinomială se bazează pe teorema de aproximare a lui Weierstrass care arată că dacă $f(x)$ este continuă pe intervalul închis $[a, b]$, atunci pentru orice $\varepsilon > 0$, există un polinom $p_n(x)$ de gradul $n = n(\varepsilon)$, astfel ca :

$$|f(x) - p_n(x)| < \varepsilon, \quad a \leq x \leq b.$$

Din nefericire criteriile existente pentru generarea polinomului de aproximare nu garantează în nici un fel că polinomul găsit este cel pus în evidență de teorema lui Weierstrass.

Un alt set de funcții liniar independente, des utilizate în teoria aproximării, sînt : $\frac{1}{2}, \cos x, \sin x, \cos 2x, \sin 2x, \dots, \cos mx, \sin mx$. În acest caz polinomul de aproximare poartă numele de *polinom trigonometric*.

$$F_m(x) = \frac{a_0}{2} + a_1 \cos x + b_1 \sin x + \dots + a_m \cos mx + b_m \sin mx = \frac{a_0}{2} + \sum_{k=1}^m (a_k \cos kx + b_k \sin kx).$$

Din expresia funcției $F_m(x)$ se observă că nu este suficientă cunoașterea funcțiilor linear independente φ_k , fiind necesară de asemenea determinarea coeficienților c_k . Pentru calculul acestor coeficienți să presupunem că spațiul M se poate organiza ca un spațiu metric, adică putem defini pe M o funcție ce măsoară distanța dintre două funcții oarecare f și g . Vom determina polinomul generalizat $F_m(x)$ (deci și coeficienții $c, c_1, \dots, c_m$) impunând condiția ca distanța dintre funcția dată f și mulțimea polinoamelor generalizate să fie cât mai mică. În funcție de modul de definire a distanței se pot pune în evidență următoarele criterii mai des utilizate.

6.2. Criteriul de aproximare prin interpolare

Dacă în spațiul metric M se definește distanța dintre două puncte f și g în felul următor: $d(f, g) = \sum_{i=0}^m |f(x_i) - g(x_i)|$, se ajunge la criteriul de aproximare prin interpolare. Dacă se înlocuiește funcția $g(x)$ prin polinomul de interpolare $F_m(x)$, condiția ca distanța dintre funcția $f(x)$ și polinomul de interpolare $F_m(x)$ să fie minimă impune sistemul de ecuații: $F_m(x_i) = f(x_i)$, $i = 0, 1, 2, \dots, n$. Dacă funcția f este dată printr-o tablă de valori:

x_k	x_0	x_1	$\dots$	x_n
$f(x_k)$	$f(x_0)$	$f(x_1)$	$\dots$	$f(x_n)$

alegînd valorile x_i ca fiind cele din tabelă, sistemul de ecuații impune de fapt coincidența dintre funcție și polinom în punctele x_i numite noduri de interpolare (fig. 6.1).

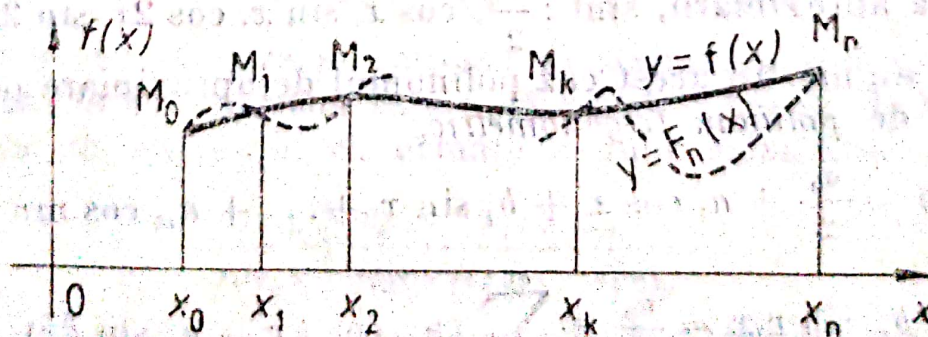


Fig. 6.1

Se observă din figură că între două noduri de interpolare abaterea polinomului de interpolare de la funcția dată $\Delta(x) = f(x) - F_m(x)$ poate atinge valori apreciabile. Coeficienții c_k ai polinomului de interpolare dat astfel:

$$F_n(x) = c_0 + c_1x + c_2x^2 + \dots + c_nx^n$$

se calculează conform celor arătate mai sus, rezolvind sistemul de $n + 1$ ecuații liniare cu $(n + 1)$ necunoscute $c_0, c_1, \dots, c_n$.

$$c_0 + c_1x_0 + c_2x_0^2 + \dots + c_nx_0^n = f(x_0)$$

$$c_0 + c_1x_1 + c_2x_1^2 + \dots + c_nx_1^n = f(x_1)$$

$$c_0 + c_1x_n + c_2x_n^2 + \dots + c_nx_n^n = f(x_n)$$

Determinantul sistemului

$$\Delta = \begin{vmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \hline 1 & x_n & x_n^2 & \dots & x_n^n \end{vmatrix}$$

este de fapt determinantul Vandermonde și acesta este diferit de zero dacă $x_i \neq x_j, i \neq j$, deci dacă nodurile de interpolare sînt distincte. Rezultă un sistem $(c_0, c_1, \dots, c_n)$ unic, deci polinomul de interpolare este unic. Schimbînd poziția nodurilor de interpolare se poate micșora valoarea abaterii maxime de la funcția dată.

Este recomandabil să se ia nodurile de interpolare dispuse analog zerourilor polinomului Cebîșev de același grad cu polinomul de aproximare.

Deci pentru polinomul de interpolare de gradul n nodurile de interpolare vor fi calculate cu formula:

$$x_k = \cos \frac{2k+1}{n} \frac{\pi}{2}, \quad k = 1, 2, \dots, n$$

Se poate pune problema dacă mărind mereu gradul polinomului de interpolare (deci făcînd pe n să tindă spre infinit) șirul corespunzător de polinoame $p_n(x)$ converge către $f(x)$.

Pentru funcții $f(x)$ relativ simple și pentru anumite alegeri ale nodurilor de interpolare (de exemplu ca fiind zerourile polinoamelor Cebîșev) convergența poate avea loc.

În general lucrînd cu noduri de interpolare egal distanțate funcția eroare $\Delta_n(x) = f(x) - p_n(x)$ are ca reprezentare grafică o curbă oscilantă, așa cum apare în fig. 6.2.

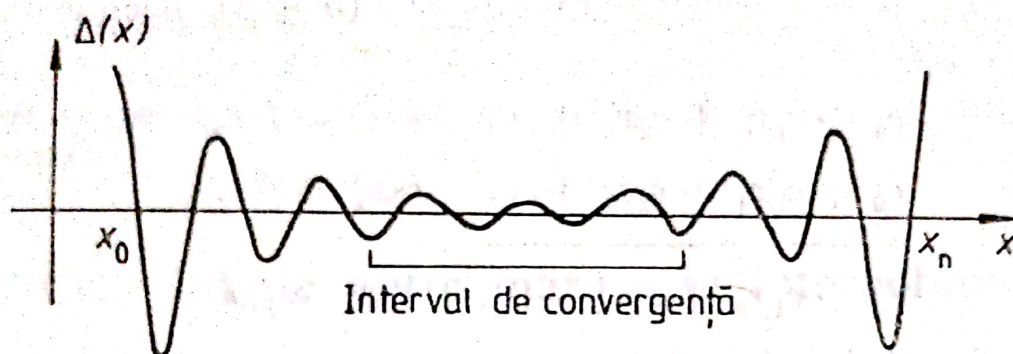


Fig. 6.2

În interiorul intervalului de convergență funcția eroare tinde spre zero atunci cînd n tinde spre infinit, dar în exteriorul acestui interval poate să crească oricît de mult.

În afara interpolării descrise mai sus cu ansambluri de noduri simple, se folosește interpolarea cu noduri multiple, în special duble. Într-un mod multiplu de interpolare de ordinul $k + 1$, se impune nu numai coincidența valorilor funcțiilor $f(x)$ cu $F_n(x)$, ci și a valorilor derivatelor lor pînă la ordinul k . Deci în cazul interpolării cu ansambluri de noduri de interpolare multiple, sistemul de ecuații ce trebuie satisfăcut de către funcția abatere este:

$$\Delta_n(x_k) = f(x_k) - F_n(x_k) = 0$$

$$\Delta'_n(x_k) = 0$$

$$\Delta''_n(x_k) = 0$$

$$\Delta_n^{(m)}(x_k) = 0$$

Fiecare nod multiplu de interpolare de ordinul $(k + 1)$ se poate considera ca o coincidență a $k + 1$ centri simpli în cazul tendinței spre zero a distanței între aceste centre. De aceea la socoteală, fiecare nod multiplu de ordinul $k + 1$

trebuie socotit drept $k + 1$ noduri simple. Geometric în cazul unui nod dublu, curba funcției abatere este tangentă la axa Ox în abscisa nodului luat în considerare. De remarcat că există o legătură între numărul punctelor în care funcția dată este tabelată și gradul polinomului de interpolare. Din această cauză o tabelă extinsă conduce la un polinom de aproximare de grad mare, ceea ce constituie un dezavantaj. Interpolarea, ce folosește tot intervalul dintre nodurile extreme ale tablei de valori, poartă numele de *interpolare globală*.

S-a observat că măbind gradul polinomului de interpolare globală, eroarea nu se micșorează întotdeauna, ba chiar adesea crește foarte mult pe unele porțiuni ale intervalului de interpolare. O soluție pentru remedierea acestei situații constă în renunțarea la interpolarea globală a funcției și trecerea la interpolarea pe porțiuni sau, altfel spus, a funcțiilor spline. O *funcție spline polinomială* de gradul $k (k \geq 1)$ relativ la diviziunea cu n noduri interioare: $a = x_0 < x_1 < x_2 < \dots < x_{n+1} = b$ este prin definiție o funcție $S(x)$ ce îndeplinește următoarele condiții:

- 1) pe fiecare subinterval $[x_i, x_{i+1}]$, $0 \leq i \leq n$, $S(x)$ coincide cu un polinom de gradul k cu coeficienți reali;
- 2) $S(x)$ este o funcție de clasa $C_{[a, b]}^{k-1}$.

Se poate verifica că mulțimea funcțiilor spline de gradul k este un spațiu vectorial de dimensiune $n + k + 1$, avînd ca bază funcțiile:

$1, x, x^2, \dots, x^k, (x - x_1)_+^k, (x - x_2)_+^k, \dots, (x - x_n)_+^k$, unde

$$(x - x_i)_+^k = \begin{cases} (x - x_i)^k & \text{dacă } x > x_i \\ 0 & \text{dacă } x \leq x_i \end{cases}$$

Rezultă că pentru determinarea unei funcții spline de gradul k se impune $n + k + 1$ condiții. În general dacă $f: [a, b] \rightarrow \mathbb{R}$ este o funcție de clasa $C_{[a, b]}^{k-1}$, atunci există o funcție spline $S(x)$ unică, de grad $2k - 1$ ce aproximează pe $f(x)$ și astfel ca următoarele $n + 2k$ condiții să fie verificate.

$$S(x_1) = f(x_1), \dots, S(x_n) = f(x_n)$$

$$S(a) = f(a), S'(a) = f'(a), \dots, S_{(a)}^{(k-1)} = f_{(a)}^{(k-1)}$$

$$S(b) = f(b), S'(b) = f'(b), \dots, S_{(b)}^{(k-1)} = f_{(b)}^{(k-1)}$$

Aplicația 1

Să se determine funcția spline de grad 1 pe intervalul $[0, \pi]$ asociată funcției $f(x) = \cos x$ și nodurilor $x_1 = \pi/3$, $x_2 = \pi/2$. În acest caz 1, x , $\left(x - \frac{\pi}{3}\right)_+$, $\left(x - \frac{\pi}{2}\right)_+$ constituie o bază și funcția $S(x)$ devine :

$$S(x) = C_1 + C_2 x + C_3 \left(x - \frac{\pi}{3}\right)_+ + C_4 \left(x - \frac{\pi}{2}\right)_+$$

Pentru determinarea celor 4 coeficienți, condițiile de mai sus devin :

$$S\left(\frac{\pi}{3}\right) = \cos \frac{\pi}{3} = \frac{1}{2}$$

$$S\left(\frac{\pi}{2}\right) = \cos \frac{\pi}{2} = 0$$

$$S(0) = \cos 0 = 1$$

$$S(\pi) = \cos \pi = -1$$

Obținem sistemul :

$$C_1 + C_2 \frac{\pi}{3} = 1$$

$$C_1 + C_2 \frac{\pi}{2} + C_3 \frac{\pi}{6} = 0$$

$$C_1 = 1$$

$$C_1 + C_2 \pi + C_3 \frac{2\pi}{3} + C_4 \frac{\pi}{2} = -1$$

Rezultă :

$$C_1 = 1, C_2 = -\frac{3}{2\pi}, C_3 = -\frac{3}{2\pi}, C_4 = \frac{1}{\pi}$$

și deci

$$\begin{aligned} \cos x \simeq S(x) &= 1 - \frac{3}{2\pi} x - \frac{3}{2\pi} \left(x - \frac{\pi}{3}\right)_+ + \\ &+ \frac{1}{\pi} \left(x - \frac{\pi}{2}\right)_+ \end{aligned}$$

Dacă în general prin interpolare se înțelege estimarea valorilor unei funcții $y(x)$ date tabelar pentru valori ale argumentului x situat între nodurile de interpolare, atunci interpolarea inversă semnifică, așa cum este de așteptat, aproximarea argumentului x ce corespunde la o valoare $y(x)$ dată, presupunând cunoscut șirul valorilor $y, y_1, \dots, y_n$. Este evident că în interpolarea inversă se schimbă rolul variabilelor x_k și y_k . În plus sîntem în situația în care noile argumente y_k nu sînt egal distanțate. Prin extrapolare se înțelege estimarea valorilor funcției $y(x)$ în puncte care se găsesc în afara intervalului în care cad argumentele date $x_0, x_1, \dots, x_n$.

6.3. Criteriul de aproximare cu abatere medie pătratică minimă

Presupunem că sîntem în spațiul $L^2[a, b]$ al funcțiilor de pătrat integrabil pe $[a, b]$. Dacă distanța dintre funcțiile f și g este dată prin

$$d(f, g) = \left\{ \frac{1}{b-a} \int_a^b \alpha(x) [f(x) - g(x)]^2 dx \right\}^{1/2},$$

atunci aproximarea ce folosește această distanță poartă numele de *aproximare cu abatere medie pătratică ponderată minimă*. Aici $\alpha(x) > 0$ este o funcție numită *pondere* (în particular $\alpha(x) = 1$).

Adeseori se definește distanța

$$d(f, g) = \left\{ \frac{1}{n+1} \sum_{i=0}^n [f(x_i) - g(x_i)]^2 \right\}^{1/2}$$

unde $x_0, x_1, \dots, x_n$ sînt $n+1$ puncte cunoscute din intervalul (a, b) . Dacă în expresia distanței de mai sus funcția $g(x)$ se înlocuiește cu $F_m(x)$, definită ca un polinom algebric de grad $m \leq n$, se ajunge tot la o aproximare cu abatere

medie pătratică minimă, metoda de aproximare purtând denumirea de *metoda celor mai mici pătrate* a lui Gauss. Această metodă este frecvent utilizată în prelucrarea matematică a datelor experimentale. Metoda celor mai mici pătrate se utilizează ori de câte ori există îndoieli privind exactitatea valorilor $f(x_i)$ culese experimental. În acest caz nu este recomandabil să se impună identificarea polinomului de aproximare cu funcția dată în valorile x_i așa cum se întâmplă în cazul interpolării. Deseori se întâmplă ca polinomul de aproximare dorit să aibă un grad mic, dar există la dispoziție un mare număr de date x_i . Deoarece criteriul de interpolare face o legătură între gradul polinomului de aproximare și numărul de valori x_i , se recomandă în acest caz metoda celor mai mici pătrate care nu impune o asemenea condiție. După cum s-a văzut, aproximarea medie pătratică cere ca polinomul de aproximare să se calculeze astfel ca valoarea abaterii

$$\Delta_{m.p.} = \sum_{i=0}^n [f(x_i) - F_m(x_i)]^2$$

să fie minimă (fig. 6.3).

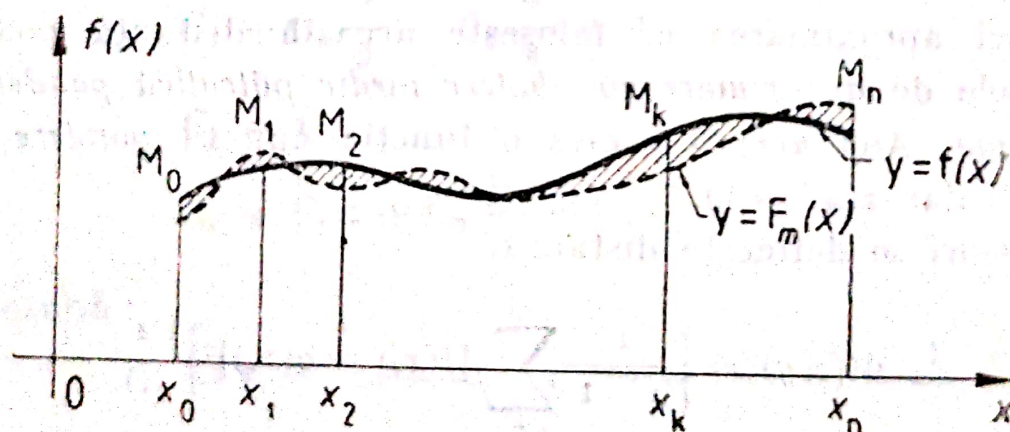


Fig. 6.3

Din punct de vedere geometric, metoda aproximării cu abateri medie pătratică minimă conduce la cerința ca aria porțiunii hașurate să fie minimă.

6.4. Criteriul de aproximare în sensul lui Cebîșev

Presupunem în acest caz că spațiul metric M este $C[a, b]$, adică mulțimea funcțiilor continue definite pe $[a, b]$, distanța dintre două funcții f și g fiind date de :

$$d(f, g) = \max_{x \in [a, b]} |f(x) - g(x)|$$

Dacă în locul funcției $g(x)$ punem polinomul de aproximare $F_m(x)$, $m \leq n$ calculul coeficienților c_k se va face din condiția ca abaterea

$$\Delta_{\minimax} = \max_{0 \leq i \leq n} |f(x_i) - F_m(x_i)|$$

să fie minimă. Aproximarea în acest caz se numește „în sensul lui Cebîșev”. Deoarece în acest caz se impune ca un maximum de funcție să fie minim, această cerință conduce prin alternanța celor două cuvinte, la o altă denumire pentru aproximarea în sensul lui Cebîșev, și anume *criteriul de minimax*, iar abaterea respectivă se numește adeseori *abatere de minimax*. Dacă valoarea minimă găsită pentru abaterea de minimax este notată cu ϵ , atunci din punct de vedere geometric acest criteriu de aproximare impune ca graficul polinomului de aproximare să nu depășească o bandă centrată pe graficul funcției $f(x)$, de lățime 2ϵ , unde ϵ are semnificația abaterii de minimax, (fig. 6.4).

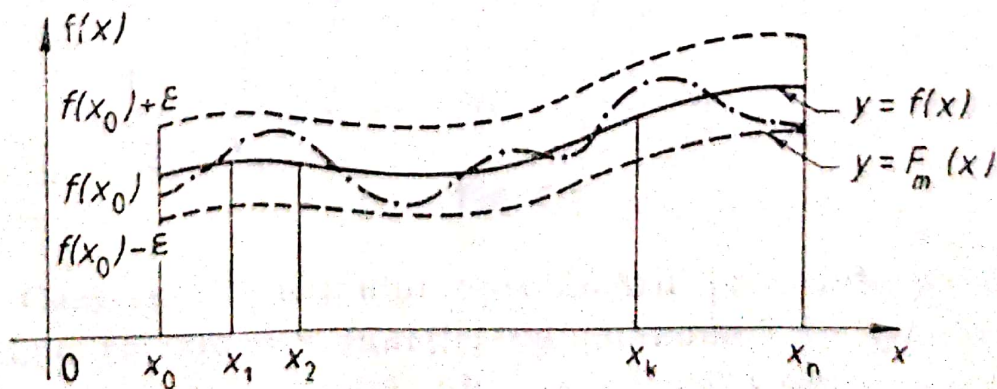


Fig. 6.4

Între cele trei criterii descrise mai sus, aproximarea prin interpolare este cea mai simplă, însă adesea precizia rezultatelor lasă de dorit. Criteriul de aproximare cu abatere medie pătratică minimă este adesea preferat în problemele de ordin tehnic, în special dacă se urmărește nu atât ordinul de mărime a erorii în fiecare punct, ci ca media erorilor să fie cât mai mică.

Criteriul de aproximare în sensul lui Cebîșev conduce în general la rezultatele cele mai precise, dar folosirea lui este deseori dificilă. Se folosește în rezolvarea problemelor care impun controlul erorii în fiecare punct, de asemenea un grad sporit de precizie a rezultatelor.

În funcție de problema dată se pot defini și alte criterii de aproximare, totuși cele trei descrise mai sus sînt în prezent cel mai des utilizate.

APROXIMAREA FUNCȚIILOR PRIN INTERPOLARE

7.1 Introducere

Interpolarea liniară a fost introdusă prima oară pentru calculul funcțiilor trigonometrice și al logaritmilor și se bazează pe aproximarea unei curbe date între două puncte, prin segmentul de dreaptă ce unește cele două puncte (fig. 7.1).

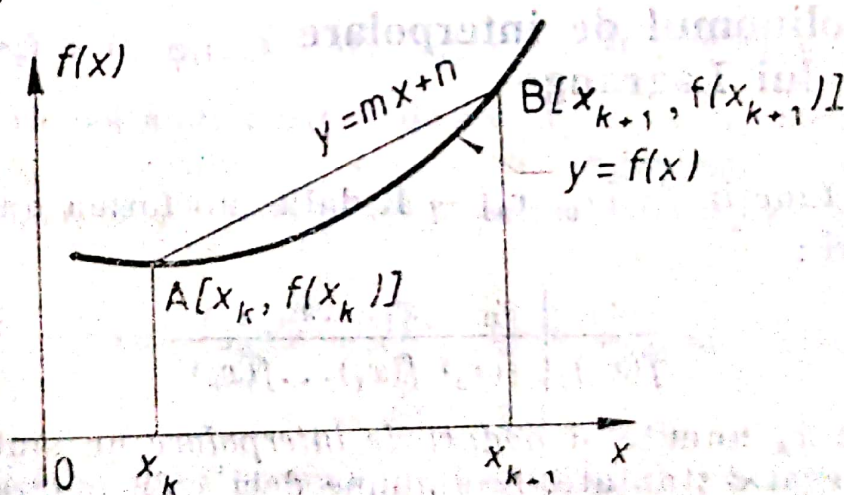


Fig. 7.1

Conform principiului aproximării prin interpolare, se impune coincidența funcției cu polinomul de gradul întâi ales pentru aproximare, (din punct de vedere geometric, o dreaptă) în nodurile de aproximare x_k, x_{k+1} .

Se scrie ecuația dreptei care trece prin punctele $A[x_k, f(x_k)]$ și $B[x_{k+1}, f(x_{k+1})]$:

$$\frac{y - f(x_k)}{f(x_{k+1}) - f(x_k)} = \frac{x - x_k}{x_{k+1} - x_k}$$

Rezultă :

$$y = \frac{f(x_{k+1}) - f(x_k)}{x_{k+1} - x_k} (x - x_k) + f(x_k),$$

deci polinomul de aproximare de gradul întâi $P(x)$ este de forma

$$P(x) = \frac{f(x_{k+1}) - f(x_k)}{x_{k+1} - x_k} x + \left[f(x_k) - \frac{f(x_{k+1}) - f(x_k)}{x_{k+1} - x_k} x_k \right]$$

Reprezentînd grafic funcția de mai sus obținem o dreaptă numită dreaptă de interpolare. Este evident că abaterea : $\Delta(x) = |f(x) - P(x)|$ depinde de intervalul $[x_k, x_{k+1}]$, de curbura funcției $f(x)$ și deci de valoarea lui $f''(x)$. Deoarece interpolarea liniară dă rareori satisfacție (eroarea este mult prea mare), se folosește de obicei un polinom de grad superior (interpolare polinomială).

7.2. Polinomul de interpolare al lui Lagrange

Fie funcția $f: [x_0, x_n] \rightarrow \mathbb{R}$ dată sub forma unei tabele de valori :

$$\begin{array}{c|c} x_k & x_0 \quad x_1 \dots x_n \\ \hline f(x_k) & f(x_0) \quad f(x_1) \dots f(x_n) \end{array}$$

Punctele x_k numite și *noduri de interpolare* nu sînt în mod necesar egal distanțate. Se impune deci coincidența funcției $f(x)$ cu un polinom de aproximare de gradul n , în nodurile de interpolare, adică :

$$f(x_k) = P_n(x_k); \quad k = 0, 1, 2, \dots, n$$

Metoda lui Lagrange propune determinarea polinomului de aproximare fără a se rezolva sistemul considerat. Pentru

Rezultă polinomul $\varphi_k(x)$ de forma :

$$\varphi_k(x) = \frac{(x-x_0)(x-x_1)\dots(x-x_{k-1})(x-x_{k+1})\dots(x-x_n)}{(x_k-x_0)(x_k-x_1)\dots(x_k-x_{k-1})(x_k-x_{k+1})\dots(x_k-x_n)}$$

iar în final polinomul de interpolare al lui Lagrange devine :

$$L_n(x) = \sum_{k=0}^n y_k \frac{(x-x_0)(x-x_1)\dots(x-x_{k-1})(x-x_{k+1})\dots(x-x_n)}{(x_k-x_0)(x_k-x_1)\dots(x_k-x_{k-1})(x_k-x_{k+1})\dots(x_k-x_n)}$$

Notînd : $a_k(x) = \prod_{\substack{i=0 \\ i \neq k}}^n \frac{(x-x_i)}{(x_k-x_i)}$; $k = 0, 1, 2, \dots, n$, avem

$$L_n(x) = \sum_{k=0}^n y_k a_k(x).$$

Polinomul de interpolare al lui Lagrange coincide cu funcția $f(x)$ în nodurile de interpoare, dar între două noduri poate să se abată mult de la valoarea funcției. Pentru a calcula funcția abatere $\epsilon_n(x) = f(x) - L_n(x)$ se procedează astfel : Fie funcția auxiliară :

$$h(x) = f(x) - L_n(x) - K(x-x_0)(x-x_1)\dots(x-x_n)$$

unde K este o constantă ce urmează a fi determinată. Se observă că $h(x_k) = 0$, pentru $k = 0, 1, 2, \dots, n$, deci, funcția $f(x)$ are $n+1$ zerouri, oricare ar fi constanta K . Alegem această constantă impunînd ca $h(x)$ să se anuleze și pentru

$$\alpha \in (x_0, x_n), \alpha \neq x_k, h(\alpha) = f(\alpha) - L_n(\alpha) - K(\alpha-x_0)(\alpha-x_1)\dots(\alpha-x_n) = 0$$

Rezultă

$$K = \frac{f(\alpha) - L_n(\alpha)}{(\alpha-x_0)(\alpha-x_1)\dots(\alpha-x_n)}$$

Dacă se socotește și rădăcina α , rezultă că funcția $h(x)$ are $n+2$ zerouri pe (x_0, x_n) ; în plus, satisface condițiile teoremei lui Rolle pe fiecare din cele $(n+1)$ subintervale care se formează folosind nodurile $x_0, x_1, \dots, x_n, \alpha$. Rezultă că

$h(x)$ are cel puțin $n + 1$ zerouri pe $[x_0, x_n]$. Raționind în mod asemănător, deducem că $h''(x)$ are cel puțin n zerouri pe $[x_0, x_n]$, iar pentru derivate de ordinul $n + 1$ există cel puțin un punct $\zeta \in (x_0, x_n)$ în care aceasta se anulează

$$h_{(\zeta)}^{(n+1)} = f_{(\zeta)}^{(n+1)} - K(n+1)! = 0.$$

Rezultă : $K = \frac{f_{(\zeta)}^{(n+1)}}{(n+1)!}.$

Dacă egalăm cele două expresii obținute pentru valoarea lui k , rezultă :

$$\epsilon_n(x) = f(x) - L_n(x) = \frac{f_{(\zeta)}^{(n+1)}}{(n+1)!} (x - x_0)(x - x_1) \dots (x - x_n)$$

Deoarece valoarea x este oarecare în intervalul $[x_0, x_n]$, $x \neq x_k$ (dar pentru $x = x_k$, evident $\epsilon_n(x) = 0$) se obține funcția eroare de forma

$$\epsilon_n(x) = \frac{f_{(\zeta)}^{(n+1)}}{(n+1)!} (x - x_0)(x - x_1) \dots (x - x_n), \quad \zeta \in (x_0, x_n)$$

Dacă funcția $f(x)$ are o derivată de ordinul $n + 1$ mărginită în valoare absolută, adică dacă există $M > 0$, astfel că oricare ar fi $x \in (x_0, x_n)$, să rezulte $|f_{(x)}^{(n+1)}| \leq M$, funcție eroare poate fi rescrisă astfel :

$$|\epsilon_n(x)| \leq \frac{M}{(n+1)!} |x - x_0| |x - x_1| \dots |x - x_n|$$

Deoarece în majoritatea cazurilor în care se utilizează polinomul de interpolare Lagrange, funcția $f(x)$ este dată printr-o tabelă de valori, derivata de ordinul $(n + 1)$ nu este cunoscută și formula anterioară este prea puțin utilă. Acesta este unul dintre dezavantajele formulei lui Lagrange, altul constind în faptul că ori de câte ori se dorește să se înlocuiască polinomul de interpolare Lagrange cu unul de grad superior, toate calculele trebuie să fie refăcute.

Aplicația 1

Să se calculeze polinomul de interpolare al lui Lagrange de gradul doi ce aproximează funcția

x_k	0	1	3
y_k	0	1	27

Avem :

$$\begin{aligned}
 L_2(x) &= 0 \frac{(x-1)(x-3)}{(0-1)(0-3)} + 1 \frac{(x-0)(x-3)}{(1-0)(1-3)} + \\
 &+ 27 \frac{(x-0)(x-1)}{(3-0)(3-1)} = 1 \frac{-x^2 + 3x}{2} + \\
 &+ 9 \frac{9x^2 - 9x}{2} = 4x^2 - 3x
 \end{aligned}$$

Polinomul Lagrange de interpolare este util și pentru realizarea interpolării inverse, deoarece acest tip de interpolare folosește tabele de valori inegal distanțate.

Aplicația 2

Pentru tabela de valori dată la aplicația 1 se cere să se determine valoarea lui x corespunzătoare valorii $y = 2$. Folosim polinomul de interpolare Lagrange de gradul doi, considerînd valorile y_k ca fiind șirul argumentelor. Avem :

$$\begin{aligned}
 L_2(y) &= x_0 \frac{(y-y_1)(y-y_2)}{(y_0-y_1)(y_0-y_2)} + x_1 \frac{(y-y_0)(y-y_2)}{(y_1-y_0)(y_1-y_2)} + \\
 &+ x_2 \frac{(y-y_0)(y-y_1)}{(y_2-y_0)(y_2-y_1)} = 1 \cdot \frac{y(y-27)}{1 \cdot (1-27)} + \\
 &+ 3 \frac{y(y-1)}{27(27-1)} = -\frac{4}{117} y^2 + \frac{121}{117} y
 \end{aligned}$$

$$\text{Rezultă : } L_2(y) = -\frac{16}{117} + \frac{242}{117} = \frac{229}{117}$$

Polinomul Lagrange poate fi extins pentru interpolare funcțiilor de două variabile. Într-adevăr, fie $(m+1)(n+1)$ puncte distincte în care este dată funcția $f(x_i, y_j)$, $i = 0, 1, \dots, m$; $j = 0, 1, \dots, n$.

Se caută un polinom $L(x, y)$ de grad cel mult m în x și de grad cel mult n în y , astfel ca :

$$L(x_i, y_j) = f(x_i, y_j), \quad i = 0, 1, \dots, m; \quad j = 0, 1, \dots, n$$

Notind :

$$a_{m,i}^{(x)} = \prod_{\substack{k=0 \\ k \neq i}}^m \frac{(x - x_k)}{(x_i - x_k)}, \quad b_{n,j}^{(y)} = \prod_{\substack{k=0 \\ k \neq j}}^n \frac{(y - y_k)}{(y_j - y_k)},$$

$$i = 0, 1, \dots, m \quad j = 0, 1, \dots, n$$

se demonstrează că polinomul Lagrange are în acest caz forma :

$$I(x, y) = \sum_{i=0}^m \sum_{j=0}^n f(x_i, y_j) a_{m,i}(x) b_{n,j}(y)$$

Aplicația 3

Se prezintă o subrutină de calcul intitulată *LAGRAN* care pe baza unei table de valori de valcarea polinomului de unei table de valori dă valoarea polinomului de interpolare Lagrange într-un punct dat. Parametrii fictivi ai subrutinei sînt :

- N* — numărul de puncte ale tablei de valori
- X* — vector conținind valorile x_k ale tablei
- Y* — vector conținind valorile y_k ale tablei
- XC* — valoare în care se cere interpolare funcției dată tabelar
- YC* — valoarea calculată pentru polinomul de interpolare Lagrange.

5 | 6 | 7

3 2 1 5	<pre> SUBROUTINE LAGRAN (N, X, Y, XC, YC) DIMENSION X(N), Y(N) YC = 0 DO 1 K = 1, N P = 1 DO 2 J = 1, N IF (K - J) 3, 2, 3 P = P * (XC - X(J)) / (X(K) - X(J)) CONTINUE YC = YC + P * Y(K) WRITE (108, 5) YC FORMAT (50 X, F 18.8) RETURN END </pre>
------------------	--

7.3. Diferențe finite

Fie funcția $y = f(x)$ tabelată în puncte echidistante și fie h pasul tabelii.

Se numește diferența finită la dreapta pentru funcția $f(x)$, expresia

$$\Delta f(x) = f(x+h) - f(x)$$

Diferențele finite de ordin superior se definesc în mod recurent :

$$\Delta^n f(x) = \Delta(\Delta^{n-1} f(x)) = \Delta^{n-1}(\Delta f(x))$$

Se verifică imediat relațiile :

$$\Delta(f+g) = \Delta f + \Delta g; \Delta(cf) = c\Delta f; \Delta^m(\Delta^n f) = \Delta^{m+n}$$

Diferențele finite de ordin superior se calculează pornind de la definiție astfel :

$$\begin{aligned} \Delta^2 f(x) &= \Delta(\Delta f(x)) = \Delta(f(x+h) - f(x)) = \\ &= f(x+2h) - 2f(x+h) + f(x) \end{aligned}$$

$$\begin{aligned} \Delta^3 f(x) &= \Delta(f(x+2h) - 2f(x+h) + f(x)) = \\ &= f(x+3h) - 3f(x+2h) + 3f(x+h) - f(x) \end{aligned}$$

Se demonstrează prin inducție completă relația

$$\Delta^n f(x) = \sum_{k=0}^n (-1)^k C_n^k f[x + (n-k)h]$$

Formulele de mai sus se simplifică dacă se calculează diferențele finite în punctele tabelii, deci pentru valorile $x = x_k$. Ținând seama că $x_k + h = x_{k+1}$ și notînd $f(x_k) = y_k$, avem :

$$\Delta y_k = y_{k+1} - y_k$$

$$\Delta^2 y_k = \Delta(\Delta y_k) = \Delta y_{k+1} - \Delta y_k$$

$$\Delta^n y_k = \Delta(\Delta^{n-1} y_k) = \Delta^{n-1} y_{k+1} - \Delta^{n-1} y_k$$

În general :

$$\Delta^n y_k = y_{n+k} - C_n^1 y_{n+k-1} + C_n^2 y_{n+k-2} - \dots + (-1)^n y_k$$

O formulă utilă, este următoarea :

$$\sum_{k=0}^{n-1} \Delta y_k = y_n - y_0$$

Ea este asemănătoare ca formă cu formula cunoscută din calcul integral :

$$\int_{x_0}^{x_n} y(x) dx = y(x_n) - y_0$$

Aplicația 4

Folosind formula de calcul

$$\sum_{k=0}^{n-1} \Delta y_k = y_n - y_0$$

să se verifice că

$$\sum_{i=1}^n i = 1 + 2 + \dots + n = \frac{n(n+1)}{2}$$

Avem nevoie de o funcție pentru care $\Delta y_i = i$ adică o problemă similară aceleia din calculul integral în care se cere determinarea funcției $y(x)$ astfel ca $dy(x) = x$.

Obținerea lui y_i cunoscând pe Δy_i poartă numele de *integrare finită*. Se verifică ușor că pentru $\Delta y = i$ rezultă y_i de forma $y_i = (i^2 - i)/2$. Deci

$$\begin{aligned} \sum_{i=1}^n i &= \sum_{i=1}^n \Delta y_i = y_{n+1} - y_1 = \frac{(n+1)^2 - (n+1)}{2} - \\ &\quad - \frac{1^2 - 1}{2} = \frac{n(n+1)}{2} . \end{aligned}$$

O altă formă utilă purtând numele de *sumare prin părți* este următoarea :

$$\sum_{i=0}^{n-1} u_i \Delta v_i = u_n v_n - u_0 v_0 - \sum_{i=0}^{n-1} v_{i+1} \Delta u_i .$$

Ea corespunde formulei de integrare prin părți a calculului integral

$$\int_{x_0}^{x_n} u(x)v'(x)dx = u(x_n)v(x_n) - u(x_0)v(x_0) - \int_{x_0}^{x_n} v(x)u'(x)dx$$

Formula de sumare prin părți se obține observând mai întâi prin calcul direct că

$$u_i \Delta v_i = \Delta(u_i v_i) - v_{i+1} \Delta u_i.$$

Sumînd de la $i = 0$ la $i = (n - 1)$ și folosind formula

$$\sum_{i=0}^{n-1} \Delta y_i = y_n - y_0,$$

rezultă

$$\sum_{i=1}^{n-1} u_i \Delta v_i = \sum_{i=1}^{n-1} \Delta(u_i v_i) - \sum_{i=1}^{n-1} v_{i+1} \Delta u_i = u_n v_n - u_0 v_0 - \sum_{i=1}^{n-1} v_{i+1} \Delta u_i.$$

Aplicația 5

Se cere să se calculeze suma seriei $\sum_{i=0}^{\infty} i a^i$; $|a| < 1$.

Observînd că $\Delta a^i = a^{i+1} - a^i = a^i (a - 1)$,

se notează $u_i = i$, $v_i = \frac{a^i}{a - 1}$.

Aplicînd formula însumării prin părți, se obține :

$$S_n = \sum_{i=0}^{n-1} i a^i = n \frac{a^n}{a - 1} - \frac{1}{a - 1} \sum_{i=0}^{n-1} a^{i+1}$$

Observînd că ultima sumă corespunde unei progresii geometrice cu rația subunitară a , avem :

$$S_n = \frac{n}{a - 1} a^n + \frac{a}{(1 - a)^2} (1 - a^n).$$

Trecînd la limită, rezultă :

$$S = \sum_{i=0}^{\infty} i a^i = \frac{a}{(1 - a)^2}.$$

În aplicații calculul diferențelor finite se face mai comod prin simple operații de scădere, pornind de la tabela de valori a funcției $f(x)$ așezată vertical. Pentru o tabelă în cinci puncte se obține :

k	x_k	y_k	Δy_k	$\Delta^2 y_k$	$\Delta^3 y_k$	$\Delta^4 y_k$
0	x_0	y_0	Δy_0	$\Delta^2 y_0$	$\Delta^3 y_0$	$\Delta^4 y_0$
1	x_1	y_1	Δy_1	$\Delta^2 y_1$	$\Delta^3 y_1$	
2	x_2	y_2	Δy_2	$\Delta^2 y_2$		
3	x_3	y_3	Δy_3			
4	x_4	y_4				

Diferențele finite de un anumit ordin se calculează scăzând diferențele finite de ordin inferior între care se găsesc, situate la stînga.

De exemplu :

Diferențele finite: $\Delta y_1 = y_2 - y_1$, $\Delta^2 y_0 = \Delta y_1 - \Delta y_0$, $\Delta^3 y_1 = \Delta^2 y_2 - \Delta^2 y_1$, etc. se numesc *diferențe finite la dreapta (progresive)*. Δy_k se calculează prin efectuarea diferențelor între ordonatele funcției $y_0, y_1, \dots, y_n$ pornind de la începutul tabelii spre dreapta. Se pot defini și *diferențe finite la stînga (regresive)* care implică pentru calculul lor formarea diferențelor între ordonatele funcției $y_0, y_1, \dots, y_n$ pornind de la sfîrșitul tabelii spre stînga. Se definesc astfel :

$$\nabla y_k = y_k - y_{k-1}$$

$$\Delta_{y_k}^2 = \nabla(\nabla y_k) = (y_k - y_{k-1}) - (y_{k-1} - y_{k-2}) =$$

$$= y_k - 2y_{k-1} + y_{k-2}$$

$$\Delta_{y_k}^n = \nabla(\nabla y_k^{n-1}) = y_n - C_k^1 y_{n-1} + C_k^2 y_{n-2} +$$

$$+ \dots + (-1)^k y_{n-k}$$

7.4. Polinomul lui Newton de interpolare

Dacă punctele de interpolare sînt echidistante, atunci pe baza diferențelor finite se poate construi polinomul de interpolare al lui Newton. Se vor folosi în cele ce urmează

diferențele finite la dreapta. Se pune deci problema determinării unui polinom de grad mai mic sau egal cu n , care să satisfacă condițiile de interpolare.

$$P_n(x_k) = y_k, \quad k = 0, 1, 2, \dots, n$$

Se ia în considerare un polinom de forma :

$$N_n^d(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \\ + \dots + c_n(x - x_0)(x - x_1) \dots (x - x_{n-1})$$

Impunind condițiile de interpolare polinomului $N_n^d(x)$ se obțin coeficienții c_k din ecuațiile :

$$c_0 = y_0$$

$$c_0 + c_1(x_1 - x_0) = y_1$$

$$c_0 + c_1(x_2 - x_0) + c_2(x_2 - x_0)(x_2 - x_1) = y_2$$

$$\dots \dots \dots$$

$$c_0 + c_1(x_n - x_0) + c_2(x_n - x_0)(x_n - x_1) + \\ + \dots + c_n(x_n - x_0) \dots (x_n - x_{n-1}) = y_n.$$

Ecuațiile de mai sus pot fi puse sub o formă mai simplă dacă se ține seama că nodurile sînt echidistante.

$$c_0 = y_0$$

$$c_0 + hc_1 = y_1$$

$$c_0 + 2hc_1 + 1 \cdot 2h^2c_2 = y_2$$

$$\dots$$

$$c_0 + nhc_1 + n(n-1)h^2c_2 + \dots + 1 \cdot 2 \dots nh^nc_n = y_n$$

Din prima ecuație rezultă coeficientul $c_0 = y_0$. Introducînd această valoare în ecuația a doua, obținem :

$$c_1 = \frac{y_1 - y_0}{h} = \frac{\Delta y_0}{h}$$

Procedind în mod asemănător rezultă și ceilalți coeficienți:

$$\begin{aligned} c_2 &= \frac{\Delta^2 y_0}{2! h^2} \\ &\vdots \\ c_n &= \frac{\Delta^n y_0}{n! h^n} \end{aligned}$$

Obținem în final polinomul lui Newton de interpolare cu diferențe finite la dreapta.

$$\begin{aligned} N_n^d(x) &= y_0 + \frac{\Delta y_0}{1! h} (x - x_0) + \frac{\Delta^2 y_0}{2! h^2} (x - x_0)(x - x_1) + \dots + \\ &+ \frac{\Delta^n y_0}{n! h^n} (x - x_0)(x - x_1) \dots (x - x_{n-1}) \end{aligned}$$

Pornind de la un polinom de interpolare de forma:

$$\begin{aligned} N_n^s(x) &= c_0 + c_1(x - x_n) + c_2(x - x_n)(x - x_{n-1}) + \dots + \\ &+ c_n(x - x_n)(x - x_{n-1}) \dots (x - x_1) \end{aligned}$$

calculele, descrise mai sus, pun în final în evidență polinomul lui Newton de interpolare cu diferențe finite la stînga:

$$\begin{aligned} N_n^s(x) &= y_n + \frac{\nabla y_n}{1! h} (x - x_n) + \frac{\nabla^2 y_n}{2! h^2} (x - x_n)(x - x_{n-1}) + \\ &+ \dots + \frac{\nabla^n y_n}{n! h^n} (x - x_n)(x - x_{n-1}) \dots (x - x_1) \end{aligned}$$

Polinomul lui Newton cu diferențe finite la dreapta se recomandă a fi utilizat cînd dorim să aproximăm valorile funcției pentru x apropiat de x_0 (începutul tabeli de valori), iar polinomul lui Newton cu diferențe finite la stînga va fi folosit pentru aproximarea valorilor funcției pentru x apropiat de x_n (sfîrșitul tabeli de valori).

Pentru a pune în evidență eroarea aproximării prin cele două polinoame de interpolare Newton să reluăm expresia erorii $\epsilon_n(x)$ pentru aproximarea polinomială, pusă în evidență în paragraful precedent.

$$\epsilon_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} (x - x_0)(x - x_1) \dots (x - x_n)$$

unde ξ se găsește între x_0 și x_n , iar x este un punct din domeniul de definiție a lui f diferit de nodurile de interpolare. Se poate demonstra relația :

$$f^{(n+1)}(x) = \lim_{h \rightarrow 0} \frac{\Delta_y^{n+1}}{h^{n+1}}$$

Renunțînd la limită se pot face aproximările :

$$f^{(n+1)}(\xi) \simeq \frac{\Delta^{n+1}y_0}{h^{n+1}} \text{ sau } f^{(n+1)}(\xi) \simeq \frac{\Delta^{n+1}y_n}{h^{n+1}}$$

Din cele de mai sus rezultă expresia erorilor $\varepsilon_n^d(x)$ și $\varepsilon_n^s(x)$ pentru polinoamele lui Newton de interpolare la dreapta, respectiv la stînga.

$$\varepsilon_n^d(x) \simeq \frac{\Delta^{n+1}y_0}{(n+1)!h^{n+1}} (x - x_0)(x - x_1) \dots (x - x_n)$$

$$\varepsilon_n^s(x) \simeq \frac{\Delta^{n+1}y_n}{(n+1)!h^{n+1}} (x - x_0)(x - x_1) \dots (x - x_n)$$

Aplicația 6

Pentru funcția dată prin următoarea tabelă de valori :

k	x_k	y_k	Δy_k	$\Delta^2 y_k$	$\Delta^3 y_k$	$\Delta^4 y_k$
0	0	-5				
1	1	1	6			
2	2	9	8	2		
3	3	25	16	8	6	
4	4	55	30	14	6	0

polinomul de interpolare cu diferențe finite la dreapta de gradul doi în punctul $x = 1,5$ dă valoarea :

$$N_2^d(1,5) = -5 + 6(1,5) + \frac{2}{2!}(1,5)(0,5) = 4,75$$

De remarcat că aceeași valoare o regăsim și dacă folosim polinomul de interpolare al lui Lagrange. Eroarea care se comite prin aproximare :

$$\epsilon_2^d(1,5) \simeq -3/8$$

Aplicația 7

Pentru funcția dată prin tabela de valori :

k	x_k	y_k	∇y_k	$\nabla^2 y_k$	$\nabla^3 y_k$	$\nabla^4 y_k$
$n-4$	0	-5				
$n-3$	1	1	6			
$n-2$	2	9	8	2		
$n-1$	3	25	16	8	6	
n	4	55	30	14	6	0

polinomul de interpolare cu diferențe finite la stînga de gradul trei în punctul $x = 3,5$ dă valoarea :

$$N_3^s(3,5) = 55 + 30(-0,5) + \frac{14}{2!}(-0,5)(0,5) + \frac{6}{3!}(-0,5)(0,5)(1,5) = 38,075.$$

Deoarece funcția tabelată $f(x) = x^3 - 2x^2 + 7x - 5$ este de forma unui polinom de gradul trei, valoarea calculată este valoarea exactă. De notat că tabela de valori de mai sus este identică cu tabela de valori de la aplicația 6, numai indicii valorilor x_k diferă.

Se pot da și diferențe finite centrale definite astfel :

$$\delta y_k = y\left(x_k + \frac{h}{2}\right) - y\left(x_k - \frac{h}{2}\right)$$

$$\delta^2 y_k = \delta(\delta y_k) = y(x_{k+1}) - 2y(x_k) + y(x_{k-1})$$

...

$$\delta^n y_k = \delta(\delta^{n-1} y_k)$$

Deoarece eroarea de interpolare tinde să fie mai mică atunci cînd se folosesc noduri de interpolare de ambele părți ale

argumentului x , rezultă că diferențele finite centrale sînt indicate în special pentru interpolare în valori ale lui x situate la mijlocul tabelii de valori.

Diferențele centrale sînt de asemenea folosite cu predicție la rezolvarea problemelor la limită.

Un tabel cu diferențe finite centrale se prezintă astfel :

k	x_k	y_k	δy_k	$\delta^2 y_k$	$\delta^3 y_k$	$\delta^4 y_k$
-2	x_{-2}	$y(x_{-2})$	$\delta y(x_{-1}-h/2)$			
-1	x_{-1}	$y(x_{-1})$	$\delta y(x_0-h/2)$	$\delta^2 y(x_{-1})$		
0	x_0	$y(x_0)$	$\delta y(x_0+h/2)$	$\delta^2 y(x_0)$	$\delta^3 y(x_0-h/2)$	
1	x_1	$y(x_1)$		$\delta^2 y(x_1)$	$\delta^3 y(x_0+h/2)$	$\delta^4 y(x_0)$
2	x_2	$y(x_2)$	$\delta y(x_1+h/2)$			

Folosind în cazul tabelii de mai sus diferențele finite situate pe linia trasată continuu se obține pentru calculul polinomului de interpolare (procedînd asemănător ca în cazul diferențelor finite la dreapta), formula lui Gauss la dreapta :

$$G_n^d(x) = y_0 + \frac{\delta y(x_0 + h/2)}{1!h} (x - x_0) + \frac{\delta^2 y(x_0)}{2!h^2} (x - x_0)(x - x_1) +$$

$$+ \frac{\delta^3 y(x_0 + h/2)}{3!h^3} (x - x_0)(x - x_1)(x - x_2) + \dots +$$

$$+ \frac{\delta^n y(x_0 + h/2)}{n!h^n} (x - x_0)(x - x_1) \dots (x - x_{n-1})$$

unde n este impar.

Dacă se utilizează diferențele finite situate pe linia trasată punctat, se obține formula lui Gauss la stînga :

$$G_n^s(x) = y_0 + \frac{\delta y(x_0 - h/2)}{1!h} (x - x_0) + \frac{\delta^2 y(x_0)}{2!h^2} (x - x_0)(x - x_1) +$$

$$+ \frac{\delta^3 y(x_0 - h/2)}{3!h^3} (x - x_0)(x - x_1)(x - x_2) + \dots +$$

$$+ \frac{\delta^n y(x_0 - h/2)}{n!h^n} (x - x_0)(x - x_1) \dots (x - x_{n-1})$$

unde n este impar.

Făcînd media aritmetică a celor două polinoame se obține formula lui Stirling. De exemplu, folosind numai diferențele finite centrale pînă la ordinul 3, se obține formula :

$$S_2(x) = y_0 + \frac{(x - x_0)}{2 \cdot 1 ! h} [\delta y(x_0 + h/2) + \delta y(x_0 - h/2)] + \\ + \frac{(x - x_0)(x - x_1)}{2 ! h^2} \delta^2 y(x_0) + \frac{(x - x_0)(x - x_1)(x - x_2)}{2 \cdot 3 ! h^3} [\delta^3 y(x_0 + h/2) + \delta^3 y(x_0 - h/2)]$$

Aplicația 8

Pentru funcția dată prin tabela de valori de mai jos să se găsească valoarea funcției în punctul $x = 2,5$.

k	x_k	y_k	δy_k	$\delta^2 y_k$	$\delta^3 y_k$	$\delta^4 y_k$
-2	0	-5	6			
-1	1	1	8	2		
0	2	9	16	8	6	
1	3	25	30	14	6	0
2	4	55				

Folosind formula lui Gauss la dreapta, polinomul de interpolare de gradul trei în punctul $x = 2,5$ dă pentru funcția tabelată valoarea exactă :

$$G_3^d(2,5) = 9 + (16)(0,5) + \frac{6(0,5)(-0,5)(1,5)}{3!} = 17,625.$$

7.5. Diferențe divizate. Polinoame de interpolare pe baza diferențelor divizate.

Polinoamele de interpolare pot fi generalizate pentru funcții reprezentate prin tabele de valori inegal distanțate (deci fără pas constant) folosind noțiunea de diferență divizată.

Diferențele divizate de la ordinul întâi la ordinul n , sînt notate și definite în felul următor :

$$y(x_0, x_1) = \frac{y_1 - y_0}{x_1 - x_0}$$

$$y(x_0, x_1, x_2) = \frac{y(x_1, x_2) - y(x_0, x_1)}{x_2 - x_0}$$

...

$$y(x_0, x_1, \dots, x_n) = \frac{y(x_1, x_2, \dots, x_n) - y(x_0, x_1, \dots, x_{n-1})}{x_n - x_0}$$

Diferențele divizate cu unu sau mai multe argumente egale, pot fi definite folosind un procedeu de trecere la limită, de exemplu :

$$y(x_0, x_0) = \lim_{x \rightarrow x_0} y(x, x_0) = \lim_{x \rightarrow x_0} \frac{y(x) - y(x_0)}{x - x_0} = y'(x_0).$$

De asemenea, diferențele divizate sînt invariante față de toate permutările argumentelor x_k cu condiția ca valorile y_k din formula de calcul să fie permutate în același mod. Aceasta reprezintă așa-numita proprietate de simetrie a diferențelor divizate. De exemplu, deoarece :

$$y(x_0, x_1) = \frac{y_0}{x_0 - x_1} + \frac{y_1}{x_1 - x_0}$$

este evident că dacă se schimbă x_0 cu x_1 și y_0 cu y_1 , atunci cei doi termeni din partea dreaptă a semnelui egal se schimbă între ei. Deci $y(x_0, x_1) = y(x_1, x_0)$. În cazul argumentelor x_k egal distanțate, diferențele divizate se reduc la diferențe finite astfel : $y(x_0, x_1, \dots, x_n) = \frac{\Delta^n y_0}{n! h^n}$.

Pentru calculul efectiv al diferențelor divizate se folosesc formulele de definiție și o tabelă organizată după modelul tabelii de calcul a diferențelor finite.

Aplicația 9

Să se calculeze diferențele divizate pentru tabela de valori :

x_k	0	1	3
y_k	1	1	5

Diferențele divizate de la ordinul întâi la ordinul n , sînt notate și definite în felul următor :

$$y(x_0, x_1) = \frac{y_1 - y_0}{x_1 - x_0}$$

$$y(x_0, x_1, x_2) = \frac{y(x_1, x_2) - y(x_0, x_1)}{x_2 - x_0}$$

...

$$y(x_0, x_1, \dots, x_n) = \frac{y(x_1, x_2, \dots, x_n) - y(x_0, x_1, \dots, x_{n-1})}{x_n - x_0}$$

Diferențele divizate cu unu sau mai multe argumente egale, pot fi definite folosind un procedeu de trecere la limită, de exemplu :

$$y(x_0, x_0) = \lim_{x \rightarrow x_0} y(x, x_0) = \lim_{x \rightarrow x_0} \frac{y(x) - y(x_0)}{x - x_0} = y'(x_0).$$

De asemenea, diferențele divizate sînt invariante față de toate permutările argumentelor x_k cu condiția ca valorile y_k din formula de calcul să fie permutate în același mod. Aceasta reprezintă așa-numita proprietate de simetrie a diferențelor divizate. De exemplu, deoarece :

$$y(x_0, x_1) = \frac{y_0}{x_0 - x_1} + \frac{y_1}{x_1 - x_0}$$

este evident că dacă se schimbă x_0 cu x_1 și y_0 cu y_1 , atunci cei doi termeni din partea dreaptă a semnelui egal se schimbă între ei. Deci $y(x_0, x_1) = y(x_1, x_0)$. În cazul argumentelor x_k egal distanțate, diferențele divizate se reduc la diferențe finite astfel : $y(x_0, x_1, \dots, x_n) = \frac{\Delta^n y_0}{n! h^n}$.

Pentru calculul efectiv al diferențelor divizate se folosesc formulele de definiție și o tabelă organizată după modelul tabelului de calcul a diferențelor finite.

Aplicația 9

Să se calculeze diferențele divizate pentru tabela de valori :

x_k	0	1	3
y_k	1	1	5

Avem :

$$y(x_0, x_1) = \frac{y_1 - y_0}{x_1 - x_0} = \frac{1 - 1}{1 - 0} = 0$$

$$y(x_1, x_2) = \frac{y_2 - y_1}{x_2 - x_1} = \frac{5 - 1}{3 - 1} = 2$$

$$y(x_0, x_1, x_2) = \frac{y(x_1, x_2) - y(x_0, x_1)}{x_2 - x_0} = \frac{2 - 0}{3} = \frac{2}{3}$$

Deci :

k	x_k	y_k	Diferențe divizate	
0	0	1		
1	1	1	0	2/3
2	3	5	2	

Folosind diferențele divizate, polinomul de interpolare al lui Newton se obține de forma :

$$N_n(x) = y_0 + (x - x_0)y(x_0, x_1) + (x - x_0)(x - x_1)y(x_0, x_1, x_2) + \dots + (x - x_0)(x - x_1) \dots (x - x_{n-1})y(x_0, x_1, \dots, x_n)$$

În formula de mai sus argumentele x_k nu trebuie să mai fie egal distanțate. Se poate arăta cu ușurință că polinomul de interpolare al lui Newton scris mai sus, reprezintă cu adevărat un polinom de interpolare, adică :

$N_n(x_k) = y_k$, $k = 0, 1, \dots, n$. Faptul că $N_n(x_0) = y_0$ este evident. Folosind definiția diferentelor divizate și proprietatea de simetrie, putem scrie în continuare :

$$y_k = y_0 + (x_k - x_0)y(x_0, x_k)$$

$$y(x_0, x_k) = y(x_0, x_1) + (x_k - x_1)y(x_0, x_1, x_k)$$

$$y(x_0, x_1, x_k) = y(x_0, x_1, x_2) + (x_k - x_2)y(x_0, x_1, x_2, x_k)$$

$$y(x_0, \dots, x_{n-2}, x_k) = y(x_0, \dots, x_{n-1}) + (x_k - x_{n-1})y(x_0, \dots, x_{n-1}, x_k).$$

Linia a doua a fost obținută astfel :

$$y(x_0, x_1, x_k) = y(x_1, x_0, x_k) = \frac{y(x_0, x_k) - y(x_1, x_0)}{x_k - x_1}$$

În continuare, linia întâi poate fi folosită pentru a verifica că $N_n(x_1) = y_1$.

Substituind linia a doua în linia întâi, se obține :

$$y_k = y_0 + (x_k - x_0)y(x_0, x_1) + (x_k - x_0)(x_k - x_1)y(x_0, x_1, x_2)$$

Sub această formă relația de mai sus poate fi folosită pentru a verifica că $N_n(x_2) = y_2$. Se continuă în același mod pentru a verifica coincidența polinomului de interpolare al lui

Newton în toate nodurile de interpolare, cu valorile y_k . Rezultatele de mai sus permit folosirea pentru polinomul de interpolare a lui Newton cu diferențe divizate a formulei erorii pentru aproximarea polinomială :

$$\varepsilon_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} (x - x_0)(x - x_1) \dots (x - x_n), \quad \xi \in (x_0, x_n)$$

Cu ajutorul diferențelor divizate, formula erorii poate fi rescrisă astfel :

$$\varepsilon_n(x) \simeq y(x, x_0, \dots, x_n)(x - x_0)(x - x_1) \dots (x - x_n)$$

Aplicația 10

Să se calculeze polinomul de interpolare de gradul doi cu diferențe divizate al lui Newton pentru tabela de valori din aplicația 9. Avem conform formulei demonstrate anterior :

$$N_2(x) = y_0 + (x - x_0)y(x_0, x_1) + (x - x_0)(x - x_1)y(x_0, x_1, x_2)$$

Folosind valorile diferențelor divizate calculate la aplicația 9 obținem :

$$\begin{aligned} N_2(x) &= 1 + (x - 0) \cdot 0 + (x - 0)(x - 1) \cdot \frac{2}{3} = \\ &= \frac{2}{3} \cdot x^2 - \frac{2}{3}x + 1 \end{aligned}$$

Linia a doua a fost obținută astfel :

$$y(x_0, x_1, x_k) = y(x_1, x_0, x_k) = \frac{y(x_0, x_k) - y(x_1, x_0)}{x_k - x_1}$$

În continuare, linia întâi poate fi folosită pentru a verifica că $N_n(x_1) = y_1$.

Substituind linia a doua în linia întâi, se obține :

$$y_k = y_0 + (x_k - x_0)y(x_0, x_1) + (x_k - x_0)(x_k - x_1)y(x_0, x_1, x_2)$$

Sub această formă relația de mai sus poate fi folosită pentru a verifica că $N_n(x_2) = y_2$. Se continuă în același mod pentru a verifica coincidența polinomului de interpolare al lui

Newton în toate nodurile de interpolare, cu valorile y_k . Rezultatele de mai sus permit folosirea pentru polinomul de interpolare a lui Newton cu diferențe divizate a formulei erorii pentru aproximarea polinomială :

$$\varepsilon_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} (x - x_0)(x - x_1) \dots (x - x_n), \quad \xi \in (x_0, x_n)$$

Cu ajutorul diferențelor divizate, formula erorii poate fi rescrisă astfel :

$$\varepsilon_n(x) \simeq y(x, x_0, \dots, x_n)(x - x_0)(x - x_1) \dots (x - x_n)$$

Aplicația 10.

Să se calculeze polinomul de interpolare de gradul doi cu diferențe divizate al lui Newton pentru tabela de valori din aplicația 9. Avem conform formulei demonstrate anterior :

$$N_2(x) = y_0 + (x - x_0)y(x_0, x_1) + (x - x_0)(x - x_1)y(x_0, x_1, x_2)$$

Folosind valorile diferențelor divizate calculate la aplicația 9 obținem :

$$\begin{aligned} N_2(x) &= 1 + (x - 0) \cdot 0 + (x - 0)(x - 1) \cdot \frac{2}{3} = \\ &= \frac{2}{3} x^2 - \frac{2}{3} x + 1 \end{aligned}$$

8. APROXIMAREA FUNCȚIILOR CU ABATERE MEDIE PĂTRATICĂ MINIMĂ

8.1. Cazul continuu

Acest tip de aproximare folosește cu predilecție sisteme de funcții ortogonale.

Fie un sistem de funcții $P_k(x)$, definite pe $[a, b]$ și aparținând spațiului vectorial L^2 al funcțiilor integrabile de pătrat integrabil. Sistemul este ortogonal dacă :

$$\int_a^b \alpha(x) P_i(x) P_j(x) dx = \begin{cases} 0, & i \neq j \\ \lambda_i > 0, & i = j \end{cases}$$

unde $\alpha(x) > 0$ este funcția pondere. În particular, $\alpha(x)$ poate fi identică cu 1. Fie funcția $f(x)$ definită și integrabilă pe intervalul $[a, b]$ și un polinom generalizat $F_m(x) = c_0 P_0(x) + c_1 P_1(x) + \dots + c_m P_m(x)$ format pe baza sistemului de funcții ortogonale. Aproximarea funcției $f(x)$ prin $F_m(x)$ cu abatere medie pătratică conduce ca integrala

$$I = \int_a^b \alpha(x) [f(x) - c_0 P_0(x) - \dots - c_m P_m(x)]^2 dx$$

să fie minimă.

Condițiile necesare de extrem impuse funcției I de $m+1$ variabile și anume $c_0, c_1, \dots, c_m$, conduc la sistemul :

$$\frac{\partial I}{\partial c_k} = -2 \int_a^b \alpha(x) [f(x) - c_0 P_0(x) - \dots -$$

$$- c_m P_m(x)] P_k(x) dx = 0, \quad k = 0, 1, \dots, m.$$

Folosind ortogonalitatea sistemului ortogonal de polinoame, sistemul de ecuații se simplifică astfel :

$$\int_a^b \alpha(x) [f(x) - c_k P_k(x)] P_k(x) dx = 0, \quad k = 0, 1, \dots, m$$

Fiecare ecuație implică o singură valoare pentru coeficienții c_k , deci :

$$c_k = \frac{\int_a^b \alpha(x) f(x) P_k(x) dx}{\int_a^b \alpha(x) P_k^2(x) dx}, \quad k = 0, 1, 2, \dots, m$$

Cu aceste valori ale coeficienților c_k , polinomul ce realizează aproximarea lui $f(x)$ cu abatere medie pătratică minimă, devine :

$$p(x) = c_0 P_0(x) + \dots + c_m P_m(x)$$

Pentru a evalua mărimea erorii care se comite prin acest tip de aproximare, să reluăm expresia integralei I :

$$I = \int_a^b \alpha(x) f^2(x) dx - 2 \sum_{k=0}^m \int_a^b \alpha(x) f(x) c_k P_k(x) dx +$$

$$+ \sum_{j,k=0}^m \int_a^b \alpha(x) c_j c_k P_j(x) P_k(x) dx$$

Al doilea termen în suma din partea dreaptă a semnului egal poate fi rescris

$$- 2 \sum_{k=0}^m c_k^2 \int_a^b \alpha(x) P_k^2(x) dx$$

ținând seama de expresia coeficientului c_k calculat. Al treilea termen în sumă se reduce, datorită ortogonalității, la valoarea

$$\sum_{k=0}^m c_k^2 \int_a^b \alpha(x) P_k^2(x) dx$$

Rezultă :

$$I = \int_a^b \alpha(x) f^2(x) dx - \sum_{k=0}^m c_k^2 \int_a^b \alpha(x) P_k^2(x) dx$$

Ultima relație arată că putem micșora abaterea medie pătratică $\Delta_{m,p} = \sqrt{\frac{1}{b-a} I}$ folosind polinoame generalizate de aproximare $F_m(x)$ formate pe baza unui număr sporit de termeni. Faptul că coeficienții c_k nu depind în nici un fel de indicele m are o mare importanță. Într-adevăr, aceasta permite calculul succesiv al coeficienților c_k oprindu-ne la acea valoare k pentru care abaterea medie pătratică se situează sub limitele admise. Atunci cînd m tinde spre infinit, se demonstrează că abaterea medie pătratică tinde către zero. Dacă nu se folosesc polinoame ortogonale, atunci o schimbare a gradului polinomului generalizat de aproximare face ca toți coeficienții să fie recalculați.

Observînd că $I \geq 0$, ultima relație de calcul permite să se pună în evidență o inegalitate remarcabilă (*inegalitatea lui Bessel*)

$$\sum_{k=0}^m c_k^2 \int_a^b \alpha(x) P_k^2(x) dx \leq \int_a^b \alpha(x) f^2(x) dx$$

Printre sistemele ortogonale de funcții, frecvent utilizate, sînt sistemul de polinoame Cebîșev și sistemul trigonometric. Fie sistemul de polinoame Cebîșev

$$\{T_n(x)\}, T_n : [-1, 1] \rightarrow \mathbb{R}, T_n(x) = \cos(n \arccos x)$$

Polinoamele Cebîșev formează un sistem ortogonal de funcții în raport cu funcția pondere $\alpha(x) = 1/\sqrt{1-x^2}$.

între-adevăr :

$$\int_{-1}^1 \frac{1}{\sqrt{1-x^2}} T_m(x) T_n(x) dx = \begin{cases} 0, & m \neq n \\ \pi/2, & m = n \neq 0 \\ \pi, & m = n = 0 \end{cases}$$

Relațiile de mai sus se verifică cu ușurință ținând cont că :

$$T_n(x) = \cos n\theta, \quad \theta = \arccos x$$

Avem :

$$\begin{aligned} I &= \int_{-1}^1 \frac{T_m(x) T_n(x)}{\sqrt{1-x^2}} dx = \int_0^\pi (\cos m\theta)(\cos n\theta) d\theta = \\ &= \frac{\sin(m+n)\theta}{2(m+n)} + \frac{\sin(m-n)\theta}{2(m-n)} \Big|_0^\pi = \end{aligned}$$

Se observă că $I = 0$ pentru $m \neq n$. Dacă $m = n = 0$ se obține valoarea π .

Dacă $m = n \neq 0$, integrala devine :

$$I = \int_0^\pi \cos^2 n\theta d\theta = \frac{1}{2} \left(\frac{\sin n\theta \cos n\theta}{n} + \theta \right) \Big|_0^\pi = \pi/2$$

Urmînd teoria generală a sistemelor ortogonale rezultă că polinomul generalizat de aproximare $F_m(x)$ format pe baza polinoamelor Cebîșev :

$$F_m(x) = c_0 T_0(x) + \dots + c_m T_m(x)$$

care face ca integrala

$$\int_{-1}^1 \frac{1}{\sqrt{1-x^2}} [f(x) - c_0 T_0(x) - \dots - c_m T_m(x)]^2 dx$$

să fie minimă, are coeficienții c_k dați de relația :

$$c_k = \frac{\int_{-1}^1 \alpha(x) f(x) T_k(x) dx}{\int_{-1}^1 \alpha(x) T_k^2(x) dx} = \frac{2}{\pi} \int_{-1}^1 \frac{f(x) T_k(x)}{\sqrt{1-x^2}} dx$$

cu excepția coeficientului c_0 , calculat astfel :

$$c_0 = \frac{1}{\pi} \int_{-1}^1 \frac{f(x)}{\sqrt{1-x^2}} dx$$

Fie sistemul trigonometric de funcții $\{\Phi_n(x)\}$

$$1, \cos \frac{\pi x}{N}, \sin \frac{\pi x}{N}, \cos \frac{2\pi x}{N}, \sin \frac{2\pi x}{N}, \dots$$

Se poate arăta că acest sistem este ortogonal pe intervalul $[-N, N]$, adică :

$$\int_{-N}^N \Phi_m(x) \Phi_n(x) dx = \begin{cases} 0, & m \neq n \\ N, & m = n \neq 0 \\ 2N, & m = n = 0 \end{cases}$$

Într-adevăr, avem :

$$\int_{-N}^N dx = 2N$$

$$\int_{-N}^N \cos \frac{k\pi x}{N} dx = \frac{N}{k\pi} \sin \frac{k\pi x}{N} \Big|_{-N}^N = 0$$

Dacă $m \neq n$,

$$\begin{aligned} \int_{-N}^N \cos \frac{m\pi x}{N} \cdot \cos \frac{n\pi x}{N} dx &= \frac{1}{2} \int_{-N}^N \left[\cos \frac{(m-n)\pi x}{N} dx + \right. \\ &\quad \left. + \cos \frac{(m+n)\pi x}{N} \right] dx = 0 \end{aligned}$$

Dacă $m = n$,

$$\int_{-N}^N \cos^2 \frac{n\pi x}{N} dx = \frac{1}{2} \int_{-N}^N \left(1 + \cos^2 \frac{n\pi x}{N} \right) dx = N$$

Asemănător se pot verifica și restul relațiilor implicate de ortogonalitate. Urmărind teoria generală, rezultă că poli-

nomul generalizat de aproximare numit și polinom trigonometric

$$\frac{a_0}{2} + \sum_{k=0}^m \left(a_k \cos \frac{k\pi x}{N} + b_k \sin \frac{k\pi x}{N} \right)$$

ce aproximează funcția $f(x)$ definită pe intervalul $[-N, N]$ periodică de perioadă $2N$, cu abatere medie pătratică minimă, are coeficienții dați de relațiile :

$$a_k = \frac{1}{N} \int_{-N}^N f(x) \cos \frac{k\pi x}{N} dx \quad k = 0, 1, 2, \dots, m$$

$$b_k = \frac{1}{N} \int_{-N}^N f(x) \sin \frac{k\pi x}{N} dx \quad k = 1, 2, \dots, m$$

Abaterea medie pătratică devine minimă odată cu integrala I . În cazul polinomului trigonometric, se obține :

$$I = \int_{-N}^N f^2(x) dx - \left[\frac{1}{2} N a_0^2 + N \sum_{k=1}^m (a_k^2 + b_k^2) \right]$$

Cînd m crește, valoarea lui I se micșorează. Se demonstrează în teoria seriei Fourier că I tinde la zero cînd m tinde spre infinit dacă funcția $f(x)$ satisface condițiile lui Dirichlet

Aplicația 1

Să se găsească polinomul trigonometric de aproximare pentru $m = 1$ în cazul funcției $y(x) = |x|$, $-\pi \leq x \leq \pi$, periodică de perioadă 2π . În cazul de față $N = \pi$. Obținem coeficienții :

$$a_0 = \frac{\pi}{2}, \quad a_1 = -\frac{4}{\pi} \cos x, \quad b_1 = 0.$$

Rezultă polinomul trigonometric

$$|x| \simeq \frac{\pi}{2} - \frac{4}{\pi} \cos x, \quad -\pi \leq x \leq \pi.$$

8.2. Cazul discret

Fie funcția $y = f(x)$ dată prin tabela de valori :

x_k	x_0	$x_1 \dots x_n$
$f(x_k)$	y_0	$y_1 \dots y_n$

Se cere să se aproximeze funcția printr-un polinom de gradul $m < n$

$$f(x) \simeq a_0 + a_1x + a_2x^2 + \dots + a_mx^m = p_m(x)$$

astfel ca abaterea medie pătratică și deci suma

$$S = \sum_{k=0}^n [y_k - (a_0 + a_1x_k + \dots + a_mx_k^m)]^2$$

să fie minimă. Această metodă imaginată mai întâi de Gauss poartă numele de *metoda celor mai mici pătrate*. Cazul cel mai simplu îl constituie aproximarea funcției $y = f(x)$ printr-un polinom de gradul întâi $ax + b$ reprezentînd geometric o dreaptă (dreapta de regresie). Se impune deci ca abaterea :

$$S(a, b) = \sum_{k=0}^n (y_k - ax_k - b)^2$$

să fie minimă.

Condițiile necesare de extrem impuse funcției de două variabile $S(a, b)$, conduce la :

$$\frac{\partial S}{\partial a} = -2 \sum_{k=0}^n x_k (y_k - ax_k - b) = 0$$

$$\frac{\partial S}{\partial b} = -2 \sum_{k=0}^n 1 \cdot (y_k - ax_k - b) = 0$$

Aceste ecuații pot fi reserise astfel :

$$(n+1)b + a \sum_{k=0}^n x_k = \sum_{k=0}^n y_k$$

$$b \sum_{k=0}^n x_k + a \sum_{k=0}^n x_k^2 = \sum_{k=0}^n x_k y_k$$

Introducînd notațiile :

$$u_0 = n+1, \quad u_1 = \sum_{k=0}^n x_k, \quad u_2 = \sum_{k=0}^n x_k^2, \quad v_0 = \sum_{k=0}^n y_k,$$

$$v_1 = \sum_{k=0}^n x_k y_k.$$

se obține soluția sistemului de mai sus de forma :

$$a = \frac{u_0 v_1 - u_1 v_0}{u_0 u_2 - u_1^2}, \quad b = \frac{u_2 v_0 - u_1 v_1}{u_0 u_2 - u_1^2}$$

Se verifică ușor că numitorul $u_0 u_2 - u_1^2$ este diferit de zero. Să calculăm derivatele parțiale de ordinul doi pentru funcția abatere

$$\frac{\partial^2 S}{\partial b^2} = 2u_0, \quad \frac{\partial^2 S}{\partial a^2} = 2u_2, \quad \frac{\partial^2 S}{\partial b \partial a} = 2u_1.$$

Se observă că primele două derivate parțiale de ordinul doi sînt pozitive. În plus, deoarece

$$\begin{aligned} \left(\frac{\partial^2 S}{\partial b \partial a} \right)^2 - \frac{\partial^2 S}{\partial b^2} \frac{\partial^2 S}{\partial a^2} &= (2u_1)^2 - 2(n+1)(2u_2) = \\ &= 4(u_1^2 - u_0 u_2) < 0 \end{aligned}$$

rezultă că soluția sistemului determină pentru funcția abatere S o valoare extremă și anume o valoare minimă.

Aplicația 2

Se dă funcția

x_k	-1	0	1	2
$f(x_k)$	1	0	1	4

Se cere dreapta de regresie ce aproximează funcția de mai sus.

Sistemul de ecuații ce determină coeficienții a și b ai dreptei de regresie devine în acest caz: $4b + 2a = 6$, $2b + 6a = 8$. Se obține soluția sistemului $a = 1$, $b = 1$, deci dreapta de regresie cerută este $y = x + 1$.

Aplicația 3

Se prezintă subrutina de calcul intitulată *REGRES*, care pe baza unei table de valori, dă ordonata drepte de regresie într-un punct dat. Parametrii fictivi ai subrutinei sint:

- N — numărul de puncte ale tablei de valori
- X — vector conținând valorile x_k ale tablei
- Y — vector conținând valorile y_k ale tablei
- XC — valoare în care se cere să se calculeze ordonata drepte
- YC — valoarea calculată a ordonatei drepte de regresie

5 | 6 | 7

```

SUBROUTINE REGRES (N, X, Y, XC, YC)
DIMENSION X (N), Y (N)
U0 = N
U1 = 0
U2 = 0
V0 = 0
V1 = 0
DO 2 I = 1, N
  U1 = U1 + X(I)
  U2 = U2 + X(I) ** 2
  V0 = V0 + Y(I)
2  V1 = V1 + X(I) * Y(I)
  A = (U0 * V1 - U1 * V0) / (U0 * U2 - U1 ** 2)
  B = (U2 * V0 - U1 * V1) / (U0 * U2 - U1 ** 2)
  YC = A * XC + B
WRITE (108, 5) YC
5  FORMAT (50 X, G 18.8)
RETURN
END

```


Se poate generaliza procedura de mai sus căutînd în loc de o funcție liniară (dreapta de regresie) o funcție polinomială de grad $m > 1$. Se cere în acest caz conform metodei celor mai mici pătrate ca funcția :

$$S = \sum_{k=0}^n (y_k - a_0 - a_1 x_k - \dots - a_m x_k^m)^2, \quad m < n$$

să fie minimă.

Condițiile necesare de extrem impun ca derivatele parțiale ale funcției abatere în raport cu coeficienții necunoscuți $a_0, a_1, \dots, a_m$ să fie zero.

$$\frac{\partial S}{\partial a_i} = -2 \sum_{k=0}^n x_k^i (y_k - a_0 - a_1 x_k - \dots - a_m x_k^m) = 0$$

Introducînd simbolurile

$$u_i = \sum_{k=0}^n x_k^i, \quad v_i = \sum_{k=0}^n y_k x_k^i$$

ecuațiile de mai sus formează sistemul :

$$u_0 a_0 + u_1 a_1 + \dots + u_m a_m = v_0$$

$$u_1 a_0 + u_2 a_1 + \dots + u_{m+1} a_m = v_1$$

$$\dots$$

$$u_m a_0 + u_{m+1} a_1 + \dots + u_{2m} a_m = v_m$$

Soluția acestui sistem determină coeficienții polinomului de aproximare.

Aplicația 4

Să se aproximeze funcția f dată prin tabela de valori calculată în 5 puncte :

x_{k-2}	x_{k-1}	x_k	x_{k+1}	x_{k+2}
y_{k-2}	y_{k-1}	y_k	y_{k+1}	y_{k+2}

conform criteriului celor mai mici pătrate, apoi să se folosească polinomul de aproximare găsit pentru a determina derivata funcției f în punctul x_k . Să facem schimbarea de variabilă $(x - x_k)/h = t$, adică $x = x_k + th$. Valorile x din

tabela de valori se obțin pentru $t = -2, -1, 0, 1, 2$. Aproximind pe f prin polinomul $a_0 + a_1 t + a_2 t^2$ se ajunge conform criteriului celor mai mici pătrate la sistemul:

$$5a_0 + 10a_2 = \sum_{j=k-2}^{k+2} y_j$$

$$10a_1 = \sum_{j=k-2}^{k+2} t_j y_j$$

$$10a_0 + 34a_2 = \sum_{j=k-2}^{k+2} t_j^2 y_j$$

Rezolvînd sistemul de ecuații de mai sus se obține:

$$a_0 = 1/35(-3y_{k-2} + 12y_{k-1} + 17y_k + 12y_{k+1} - 3y_{k+2})$$

$$a_1 = 1/10(-2y_{k-2} - y_{k-1} + y_{k+1} + 2y_{k+2})$$

$$a_2 = 1/14(2y_{k-2} - y_{k-1} - 2y_k - y_{k+1} + 2y_{k+2})$$

Deoarece

$$f'(x_k) \simeq P'(x_k) = a_1/h$$

se obține în final

$$f'(x_k) \simeq \frac{1}{10h}(-2y_{k-2} - y_{k-1} + y_{k+1} + 2y_{k+2})$$

Sistemul de ecuații care conduce la determinarea coeficienților polinomului de aproximare se rezolvă cu ușurință pentru valori mici ale lui m , așa cum s-a văzut, dar pentru valori mari de lui m devine adeseori nestabil.

Această dificultate poate fi evitată lucrînd cu sisteme de funcții ortogonale.

Se poate arăta astfel că dacă se folosește o tabelă de valori

x	0	1...2N
$y(x)$	$y(0)$	$y(1)...y(2N)$

calculată într-un număr impar de puncte $2N + 1$, egal distanțate, atunci polinomul trigonometric de aproximare (prin metoda celor mai mici pătrate) pentru funcția $y(x)$ dată tabelar și periodică de perioada $2N$ este de forma

$$y(x) \simeq T_{2N+1}(x) = \frac{a_0}{2} + \sum_{k=1}^N \left(a_k \cos \frac{2k\pi x}{2N+1} + b_k \sin \frac{2k\pi x}{2N+1} \right)$$

coeficienții a_k și b_k fiind calculați cu formulele :

$$a_k = \frac{2}{2N+1} \sum_{x=0}^{2N} y(x) \cos \frac{2k\pi x}{2N+1}, \quad k = 0, 1, 2, \dots, N$$

$$b_k = \frac{2}{2N+1} \sum_{x=0}^{2N} y(x) \sin \frac{2k\pi x}{2N+1}, \quad k = 1, 2, \dots, N$$

Într-adevăr, să luăm un polinom trigonometric de aproximare oarecare, dar avînd forma :

$$T_{2M+1}^{(x)} = \frac{A_0}{2} + \sum_{k=1}^M \left(A_k \cos \frac{2k\pi x}{2N+1} + B_k \sin \frac{2k\pi x}{2N+1} \right), \quad M < N$$

și să arătăm că abaterea

$$S = \sum_{x=0}^{2N} [y(x) - T_{2M+1}^{(x)}]^2$$

devine minimă atunci cînd coeficienții $A_k = a_k$ și $B_k = b_k$. Termenul general al abaterii este de forma :

$$y(x) - T_{2M+1}^{(x)} = \frac{a_0 - A_0}{2} + \sum_{k=1}^M \left[(a_k - A_k \cos \frac{2k\pi x}{2N+1} + b_k \sin \frac{2k\pi x}{2N+1} - B_k \sin \frac{2k\pi x}{2N+1}) \right]$$

$$- A_k) \cos \frac{2k\pi x}{2N+1} + (b_k - B_k) \sin \frac{2k\pi x}{2N+1} \Big] +$$

$$+ \sum_{k=M+1}^N \left(a_k \cos \frac{2k\pi x}{2N+1} + b_k \sin \frac{2k\pi x}{2N+1} \right)$$

Ridicînd la pătrat și folosind proprietatea de ortogonalitate a sistemului finit de funcții trigonometrice $\left\{ \cos \frac{2k\pi x}{2N+1}, \sin \frac{2k\pi x}{2N+1} \right\}$;

$$\sum_{x=0}^{2N} \sin \frac{2i\pi x}{2N+1} \sin \frac{2k\pi x}{2N+1} = \begin{cases} 0, & i \neq k \text{ sau } i = k = 0 \\ (2N+1)/2, & i = k \neq 0 \end{cases}$$

$$\sum_{x=0}^{2N} \cos \frac{2i\pi x}{2N+1} \cos \frac{2k\pi x}{2N+1} = \begin{cases} 0, & i \neq k \\ (2N+1)/2, & i = k \neq 0 \\ N+1, & i = k = 0 \end{cases}$$

$$\sum_{x=0}^{2N} \sin \frac{2i\pi x}{2N+1} \cos \frac{2k\pi x}{2N+1} = 0$$

se obține :

$$S = \frac{2N+1}{4} (a_0 - A_0)^2 + \frac{2N+1}{2} \sum_{k=1}^M [(a_k - A_k)^2 + (b_k - B_k)^2] + \frac{2N+1}{2} \sum_{k=M+1}^N a_k^2 + b_k^2$$

Evident pentru a micșora abaterea S , trebuie să luăm $A_0 = a_0$, $A_k = a_k$, $B_k = b_k$.

Se poate arăta că abaterea S descrește atunci cînd M crește atîngînd valoarea 0 pentru $M = N$, cu alte cuvinte în acest ultim caz polinomul trigonometric de aproximare coincide cu un polinom generalizat de interpolare. De multe ori se

prefera să se folosească pentru funcția $y(x)$ periodică de perioadă $2N$, o tabelă de valori de forma

x	0	1	...	$2N - 1$
$y(x)$	$y(0)$	$y(1)$	...	$y(2N - 1)$

dată într-un număr par de puncte $2N$. Atunci polinomul trigonometric este de forma :

$$f(x) \simeq T_{2N}(x) = \frac{a_0}{2} + \sum_{k=1}^{N-1} \left(a_k \cos \frac{k\pi x}{N} + b_k \sin \frac{k\pi x}{N} \right) + \frac{1}{2} a_N \cos \pi x$$

unde coeficienții a_k , b_k se calculează cu formulele :

$$a_k = \frac{1}{N} \sum_{x=0}^{2N-1} y(x) \cos \frac{\pi k x}{N}, \quad k = 0, 1, \dots, N$$

$$b_k = \frac{1}{N} \sum_{x=0}^{2N-1} y(x) \sin \frac{\pi k x}{N}, \quad k = 1, 2, \dots, N - 1$$

Aplicația 5

Să se determine un polinom trigonometric de aproximare pentru funcția $y(x)$ dată prin tabela de valori

x	0	1	2	3
$y(x)$	0	1	4	9

Avem $2N - 1 = 3$ deci $N = 2$.

Polinomul trigonometric va fi de forma :

$$T_4(x) = \frac{a_0}{2} + a_1 \cos \frac{\pi x}{2} + b_1 \sin \frac{\pi x}{2} + \frac{1}{2} a_2 \cos \pi x$$

Coeficienții a_k , b_k calculați din formulele de mai sus sînt :

$$a_0 = 7, \quad a_1 = -2, \quad b_1 = -4, \quad a_2 = -3$$

Rezultă :

$$T_1(x) = \frac{7}{2} - 2 \cos \frac{\pi x}{2} - 4 \sin \frac{\pi x}{2} - \frac{3}{2} \cos \pi x$$

Frecvent intervin în tehnică funcții periodice de perioadă 2π definite printr-o tabelă de valori dată în n puncte ce împart intervalul $[0, 2\pi]$ în $n - 1$ părți egale.

φ_k	$0 = \varphi_1, \varphi_2, \dots, \varphi_k = (k - 1) \frac{2\pi}{n - 1}, \dots, \varphi_n = 2\pi$				
$f(\varphi_k)$	$f_1,$	f_2	$\dots$	f_k	$\dots$
					f_n

De fapt cazul în care o funcție $g(x)$ este definită pe intervalul $[\alpha, \alpha + P]$ poate fi redus la cazul precedent, printr-o schimbare de variabilă de forma :

$$x = \alpha + \frac{P}{2\pi} \varphi$$

Într-adevăr, această substituție transformă funcția $g\left(\alpha + \frac{P}{2\pi} \varphi\right)$ într-o funcție $f(\varphi)$ definită pe intervalul $[0, 2\pi]$. Dacă tabela este dată într-un număr par de puncte, adică $n = 2N$ atunci polinomul trigonometric de aproximare pus în evidență în acest caz, apare sub o formă simplificată :

$$f(\varphi) \simeq \frac{a_0}{2} + \sum_{k=1}^{N-1} (a_k \cos k\varphi + b_k \sin k\varphi) + \frac{1}{2} a_N \cos N\varphi$$

Coeficienții a_k și b_k se calculează cu formulele :

$$a_k = \frac{2}{n} \sum_{i=1}^n f_i \cos k\varphi_i, \quad k = 0, 1, 2, \dots, N$$

$$b_k = \frac{2}{n} \sum_{i=1}^n f_i \sin k\varphi_i, \quad k = 1, 2, \dots, N - 1$$

Dacă funcția $f(\varphi)$ se dă printr-o tabelă de numai 12 valori, adică $n = 12$, apar mari simplificări în calculul coeficienților Fourier cu relațiile scrise anterior. Într-adevăr, în calcule

nu mai figurează decît 21 de funcții trigonometrice care cu excepția semnului, nu au decît trei valori distincte diferite de zero : 0,5, $\sqrt{3}/2 = 0,866$, 1.

În acest caz calculul coeficienților Fourier este indicat să se facă prin metoda Runge folosind tabele tip. Calculul se va desfășura în felul următor : se scriu cele 12 valori f_i ale funcției f , astfel :

	f_{12}	f_{11}	f_{10}	f_9	f_8	f_7	f_6
Sume :	S_0	S_1	S_2	S_3	S_4	S_5	S_6
Diferențe :		d_1	d_2	d_3	d_4	d_5	

Sumele S_k se calculează în modul următor : $S_0 = f_{12}$, $S_6 = f_6$, în rest S_k reprezintă suma valorilor funcției f aflate pe coloana k . Diferențele d_k se formează luînd de asemenea în considerare valorile funcției aflate pe coloana k în tabelă. De exemplu $d_3 = f_3 - f_9$. Se calculează în continuare în mod analog sumele σ_k și θ_k respectiv diferențele δ_k și η_k folosind tabelele

	S_0	S_1	S_2	S_3		d_1	d_2	d_3
	S_6	S_5	S_4			d_5	d_4	
Sume :	σ_0	σ_1	σ_2	σ_3	Sume :	θ_1	θ_2	θ_3
Diferențe :	δ_0	δ_1	δ_2		Diferențe :	η_1	η_2	

Calculul coeficienților Fourier se face în continuare utilizînd următoarele două tabele :

Coeficienții sinusurilor							
0,5	θ_1						
0,866		θ_2	η_1	η_2			
1	θ_3				θ_1	θ_3	
Sumele	I	II	I	II	I	II	
I + II		$6b_1$		$6b_2$			
I - II		$6b_5$		$6b_4$		$6b_3$	

Coeficienții cosinurilor

0,5	δ_2		$-\sigma_2$		σ_1			
0,866	δ_1							
1	$\sigma_0 + \sigma_2$	$\sigma_1 + \sigma_3$	δ_0	δ_1	$-\sigma_3$	σ_3	δ_0	δ_2
Sumele	I	II	I _a	II	I	II	I	II
I + II	$12a_0$		$6a_1$		$6a_2$			
I - II	$12a_6$		$6a_5$		$6a_4$		$6a_3$	

În cele două tabele cifrele 0,500, 0,866 și 1,000 indică faptul că trebuie înmulțite prin aceste valori corespunzătoare aflate pe linia ce corespunde cifrei respective în coloanele din partea dreaptă. Se calculează în continuare sumele I și II ca în tabelele de sume. În final rezultă coeficienții a_k și b_k luând în considerare valorile respective I + II, de asemenea I - II. Se pot scrie, de exemplu, folosind tabelele de mai sus :

$$6b_1 = 0,5 \cdot 0_1 + 1 \cdot 0_3 - 0,866 \cdot 0_2$$

$$6b_3 = 1 \cdot 0_1 - 1 \cdot 0_3$$

$$6a_2 = -0,5 \sigma_2 - 1 \cdot \sigma_3 + 0,5 \sigma_1 + 1 \cdot \sigma_3$$

Pornind de la sistemul de polinoame Cebîșev $\{T_n(x)\}$ se poate forma polinomul de aproximare pentru funcția $y(x)$ dată prin tabela de valori :

$$\frac{x_k}{y_k} \left| \frac{x_0 x_1 \dots x_{N-1}}{y_0 y_1 \dots y_{N-1}} \right., \text{ unde } x_k = [(2k+1)\pi/(2N)]$$

$$k = 0, 1, \dots, N-1$$

de forma : $P(x) = c_0 T_0(x) + c_1 T_1(x) + \dots + c_m T_m(x)$.

Se observă că valorile x_k sînt alese ca fiind zerourile polinomului Cebîșev $T_N(x)$. Coeficienții c_k se calculează folosind proprietatea de ortogonalitate pentru sistemele finite de polinoame Cebîșev.

$$\sum_{k=0}^{N-1} T_i(x_k) T_j(x_k) = \begin{cases} 0, & i \neq j \\ N/2, & i = j \neq 0 \\ N, & i = j = 0 \end{cases}$$

Coeficienții se obțin de forma :

$$c_0 = \frac{1}{N} \sum_{i=1}^{N-1} y_i, \quad c_k = \frac{2}{N} \sum_{i=1}^{N-1} y_i T_k(x_i), \quad k = 1, 2, \dots, m.$$

9. APROXIMAREA FUNCȚIILOR ÎN SENSUL LUI CEBÎȘEV

9.1. Cazul continuu

Polinomul de gradul n poate fi considerat un polinom generalizat format pe baza monoamelor $1, x, x^2, \dots, x^n$. Examinînd comportarea acestor monoame pe intervalul $[-1, 1]$, se observă că fiecare din ele își atinge valoarea maximă 1, la $x = \pm 1$ și valoarea minimă 0, la $x = 0$, cu alte cuvinte valorile extreme sînt atinse la capetele și la mijlocul intervalului. Dacă funcția $f(x)$ este aproximativă prin polinomul: $f(x) \simeq a + a_1x + a_2x^2 + \dots + a_nx^n$, modificarea coeficienților sau schimbarea gradului polinomului, operații care se execută în procesul de aproximare, vor produce erori mici pentru valori ale lui x situate în apropierea lui zero și erori mari pentru valori ale lui x situate în apropierea lui ± 1 . Deci, cu toate avantajele care provin din expresia lor simplă, setul de monoame are dezavantajul unor extreme neuniform distribuite. Mult mai potrivite par din acest punct de vedere funcțiile trigonometrice: $\cos 0, \cos 2\theta, \dots, \cos n\theta$, așa cum se observă din fig. 9.1. Faptul că aceste funcții sînt definite pe intervalul $[0, \pi]$ nu constituie o piedică, știut fiind că putem transforma un interval oarecare $[a, b]$ în intervalul $[-1, 1]$ prin schimbarea de variabilă

$$x = \frac{2y - b - a}{b - a}, \quad y \in [a, b], \quad x \in [-1, 1].$$

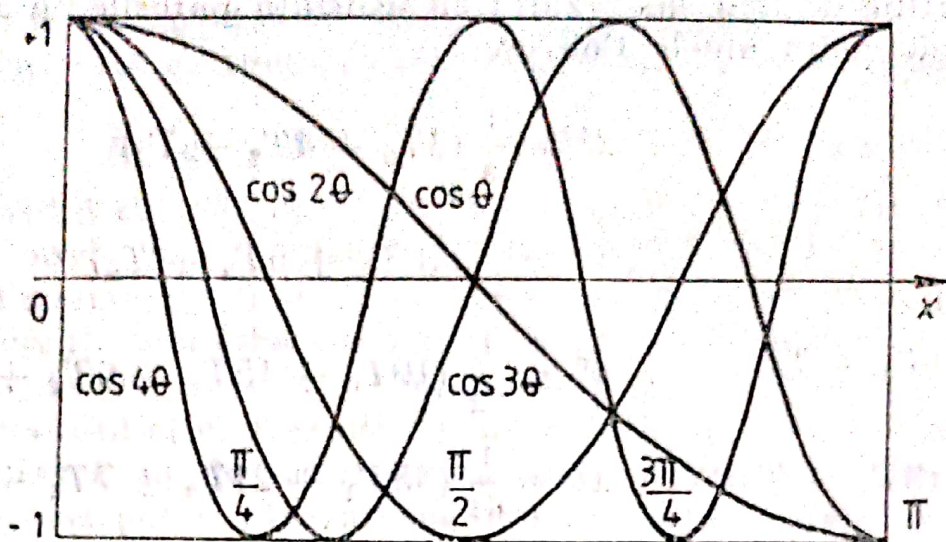


Fig. 9.1

Dificultatea utilizării funcțiilor $\cos n\theta$ în aproximarea funcțiilor constă în faptul că sînt dificil de evaluat numeric. Din această cauză se preferă transformarea funcțiilor $\cos n\theta$ definite pe intervalul $[-\pi, \pi]$ într-un set de polinoame definite pe intervalul $[-1, 1]$, folosind transformarea: $\theta = \arccos x$.

Obținem:

$$T_1(x) = \cos 0 = \cos(\arccos x) = x$$

$$T_2(x) = \cos 2\theta = \cos(2 \arccos x) = 2 \cos^2(\arccos x) - 1 = 2x^2 - 1$$

$$T_n(x) = \cos n\theta$$

Pentru calculul polinoamelor $T_n(x)$, numite *polinoame Cebîșev*, se folosește identitatea trigonometrică

$$\cos n\theta = 2 \cos \theta \cdot \cos (n-1)\theta - \cos (n-2)\theta$$

care conduce la relația de recurență:

$$T_n(x) = 2x T_{n-1}(x) - T_{n-2}(x).$$

Primele opt polinoame Cebîșev sînt următoarele:

$$T_0 = 1$$

$$T_4 = 8x^4 - 8x^2 + 1$$

$$T_1 = x$$

$$T_5 = 16x^5 - 20x^3 + 5x$$

$$T_2 = 2x^2 - 1$$

$$T_6 = 32x^6 - 48x^4 + 18x^2 - 1$$

$$T_3 = 4x^3 - 3x$$

$$T_7 = 64x^7 - 112x^5 + 56x^3 - 7x$$

Din relațiile de mai sus rezultă cu ușurință puterile lui x în funcție de polinoamele Cebîșev.

$$\begin{aligned} 1 &= T_0 & x^4 &= \frac{1}{8} (3T_0 + 4T_2 + T_4) \\ x &= T_1 & x^5 &= \frac{1}{16} (10T_1 + 5T_3 + T_5) \\ x^2 &= \frac{1}{2} (T_0 + T_2) & x^6 &= \frac{1}{32} (10T_0 + 15T_2 + 6T_4 + T_6) \\ x^3 &= \frac{1}{4} (3T_1 + T_3) & x^7 &= \frac{1}{64} (35T_1 + 21T_3 + 7T_5 + T_7) \end{aligned}$$

Folosind definiția, rezultă că $T_n(x)$ este zero pentru $0 = (2k + 1)\pi/2n$, deci rădăcinile polinomului sînt :

$$x_k = \cos \frac{(2k+1)\pi}{2n}, \quad k = 1, \dots, n.$$

Cele n valori de mai sus sînt cuprinse între -1 și 1 . Ținînd seama că $T_n(x)$ are numai n zerouri, rezultă că toate sînt cuprinse în intervalul $[-1, 1]$. Deoarece polinomul $T_n(x)$ este egal cu un cosinus, rezultă că în intervalul $[-1, 1]$ nu poate să depășească valoarea 1 , atingînd aceste valori limită în $(n+1)$ puncte.

$T_n(x) = (-1)^k$, în punctele $\lambda_k = \cos k\pi/n$, $k = 1, 1, \dots, n$.

Ansamblul punctelor λ_k formează suportul lui Cebîșev. O proprietate utilă a polinoamelor Cebîșev o constituie faptul că dintre toate polinoamele $p_n(x)$ de gradul n care au coeficientul lui x^n egale cu 1 , cel pentru care abaterea sa de la zero este minimă în intervalul $[-1, 1]$ este polinomul :

$$2^{1-n} T_n(x).$$

Într-adevăr, să observăm că deoarece valoarea maximă absolută pentru $T_n(x)$ este 1 , atunci pentru $2^{1-n} T_n(x)$ această valoare este 2^{1-n} ; în plus, polinomul $2^{1-n} T_n(x)$ are coeficientul termenului x^n egal cu 1 . Să presupunem acum că ar exista un polinom $p_n(x)$ avînd coeficientul lui x^n egal cu 1 , astfel ca $|p_n(x)| < 2^{1-n}$, pentru orice $x \in [-1, 1]$. Atunci

datorită faptului că polinomul $2^{1-n}T_n(x)$ ia valori extreme alternante ca semn în cele n puncte, $\lambda_k = \cos k\pi/n$, avem :

$$p_n(\lambda_0) < 2^{1-n}T_n(\lambda_0), \quad p_n(\lambda_1) > 2^{1-n}T_n(\lambda_1), \text{ etc.}$$

Rezultă că polinomul diferență $p_n(x) - 2^{1-n}T_n(x)$ are gradul cel mult $n - 1$, deoarece coeficienții lui x^n în cele două polinoame $p_n(x)$ și $2^{1-n}T_n(x)$ sînt egali, în plus polinomul diferență schimbă semnul de n ori în fiecare d'n cele n intervale $(\lambda_{k-1}, \lambda_k)$, $k = 0, 1, 2, \dots, n - 1$. Se ajunge deci la o contradicție. Rezultă că nu există un polinom $p(x)$ avînd abaterea sa de la zero mai mică decît $2^{1-n}T_n(x)$. Cele de mai sus pot fi utilizate pentru micșorarea erorii ce apare în aproximarea unei funcții printr-un polinom de interpolare de gradul n . În acest caz, eroarea este :

$$\varepsilon_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} (x - x_0)(x - x_1) \dots (x - x_n).$$

Singura cale efectivă de a micșora eroarea este de a minimiza valoarea maximă a polinomului de gradul $(n + 1)$ care apare în partea dreaptă a relației anterioare. Acest lucru implică alegerea nodurilor de interpolare $x_0, x_1, \dots, x_n$ ca fiind zerourile polinomului Cebîșev $T_{n+1}(x)$, adică :

$$x_k = \cos \frac{(2k+1)\pi}{2n+2}, \quad k = 0, 1, \dots, n.$$

Dacă se lucrează pe un interval arbitrar $a \leq y \leq b$, deci se trece de la intervalul $[-1, 1]$ la intervalul $[a, b]$, nodurile de interpolare y_k se obțin d'n valorile x_k prin relația de transformare

$$y_k = \frac{x_k(b-a) + (b+a)}{2}.$$

Teoria polinoamelor Cebîșev poate fi generalizată. Se demonstrează astfel că dacă funcția $f(x)$ este continuă pe intervalul $[a, b]$, există un polinom $P(x)$ unic, de grad n numit *polinom de minimar*, astfel că abaterea :

$$\Delta_{\min \max} = \max_{x \in [a, b]} |f(x) - P_n(x)|$$

este minimă. În plus, există cel puțin $(n + 2)$ puncte în care Δ minimax atinge valoarea sa maximă, alternând semnul acesteia. Această observație deschide și calea pentru calculul polinomului de minimax de gradul n .

$$P_n(x) = c_0 + c_1x + c_2x^2 + \dots + c_nx^n.$$

Într-adevăr, fie $x_0, x_1, \dots, x_{n+1}$ cele $n + 2$ puncte în care funcția abatere atinge valorile sale limită $\pm M$. Se poate scrie sistemul de $n + 2$ ecuații :

$$f(x_0) - P_n(x_0) = \varepsilon M$$

$$f(x_1) - P_n(x_1) = -\varepsilon M$$

$$f(x_{n+1}) - P_n(x_{n+1}) = (-1)^n \varepsilon M, \text{ unde } \varepsilon = +1 \text{ sau } -1.$$

În plus, deoarece valorile $x_0, x_1, x_2, \dots, x_n$ corespund punctelor în care funcția abatere ia valoare extremă, derivata sa în aceste puncte trebuie să fie egală cu zero.

$$f'(x_0) - P'_n(x_0) = 0$$

$$f'(x_1) - P'_n(x_1) = 0$$

$$f'(x_{n+1}) - P'_n(x_{n+1}) = 0$$

Punînd împreună relațiile de mai sus, obținem un sistem de $2n + 4$ ecuații ce poate fi scris prescurtat astfel :

$$f(x_k) - P_n(x_k) = (-1)^k \varepsilon M$$

$$f'(x_k) - P'_n(x_k) = 0, \quad k = 0, 1, \dots, n+1$$

$$\varepsilon = +1 \text{ sau } -1.$$

Sistemul celor $2n + 4$ ecuații conține $2n + 4$ necunoscute și anume :

$$c_0, c_1, \dots, c_n, x_0, x_1, \dots, x_{n+1}, M.$$

Aplicația 1

Să se determine polinomul de minimax de gradul 1 care aproximează funcția $f(x)$ cu $f''(x) > 0$, pe intervalul $[a, b]$. Să presupunem polinomul de minimax căutat, de

forma $P(x) = mx + n$. Conform teoriei, trebuie să găsim 3 puncte $x_1 < x_2 < x_3$ în $[a, b]$ pentru care funcția abatere $\Delta(x) = f(x) - P(x)$ atinge valorile sale extreme alternând ca semn. Deoarece $\Delta'(x_2) = f'(x_2) - P'(x_2) = f'(x_2) - m = 7$, rezultă $f'(x_2) = m$. Dar $f''(x) > 0$, deci f' este strict crescătoare și poate să fie egală cu m doar o singură dată. Rezultă că în interiorul intervalului se găsește doar punctul x_2 ; x_1 și x_3 sînt situate la capetele intervalului, $x_1 = a$, $x_3 = b$. Rezultă sistemul de ecuații:

$$f(a) - P(a) = -[f(x_2) - P(x_2)] = f(b) - P(b) = M,$$

unde M reprezintă abaterea maximă.

Rezolvînd acest sistem, obținem:

$$m = \frac{f(b) - f(a)}{b - a}, \quad n = \frac{f(a) + f(x_2)}{2} - \frac{(a + x_2)[f(b) - f(a)]}{2(b - a)},$$

unde punctul x_2 este determinat din relația

$$f'(x_2) = \frac{f(b) - f(a)}{b - a}.$$

De exemplu, pentru funcția $f(x) = x^2$, $x \in [0, 2]$ se obține:

$$f'(x_2) = 2x_2 = 2, \text{ deci } x_2 = 1.$$

În continuare, folosind formulele de mai sus, rezultă

$$m = 2, \quad n = -1/2.$$

În concluzie, polinomul de minimax de gradul întâi este $P(x) = 2x - 1/2$, avînd reprezentarea grafică dată în figura 9.2. Se observă că graficul polinomului de minimax nu depășește o bandă trasată centrat pe graficul funcției de lățime

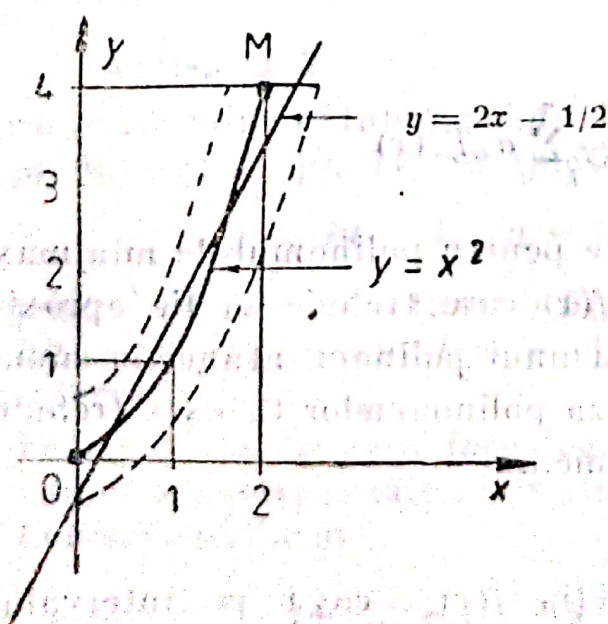


Fig. 9.2

egală cu $2M = 1$. Valoarea maximă a abaterii $M = 1/2$ se obține în punctul $x = 0$.

Se poate compara pe fig. 9.2 calitatea aproximării prin dreapta de minimax în comparație cu dreapta de interpolare $y = 2x$ trasată în punctele de coordonate $(0, 0)$ și $(2, 4)$ sau cu dreapta de regresie care în cazul acesta este $y = 2x - 1/3$.

Dacă funcția $f(x)$ este continuă, rezultă și $\Delta_{\min n, x}$ de asemenea funcție continuă și deoarece ia valori egale și opuse ca semn în $n + 2$ puncte, se va anula de asemenea în $n + 2$ puncte. Rezultă că pe intervalul (a, b) există $n + 1$ puncte în care funcția $f(x)$ coincide cu polinomul de minimax, deci acest polinom este în realitate un polinom de interpolare de gradul n , nodurile de interpolare fiind situate între punctele de extrem. În acest caz ne putem folosi de formula erorii pentru interpolare. Dacă derivata $f^{(n+1)}(x)$ există și este constantă, polinomul de minimax de gradul n poate fi construit ca un polinom de interpolare folosind o bază de noduri de interpolare dată de zerourile polinomului Cebîșev de gradul n . Calculul exact al polinomului de minimax de grad superior lui unu este d'ficil.

Se poate însă arăta că dacă funcția $f: [-1, 1] \rightarrow \mathbb{R}$ poate fi dezvoltată într-o serie formată pe baza polinoamelor Cebîșev

$$f(x) = \sum_{k=0}^{\infty} a_k T_k(x),$$

atunci o sumă parțială

$$p_n(x) = \sum_{k=0}^n a_k T_k(x)$$

reprezintă o bună aproximare pentru polinomul de minimax de gradul n . Dacă funcția $f(x)$ care trebuie să fie aproximată este ea însăși de forma unui polinom, atunci în suma parțială $p_n(x)$ formată pe baza polinoamelor Cebîșev trebuie să renunțăm la ultimul termen.

Aplicația 2

Să se aproximeze funcția $f(x) = \cos x$ pe intervalul $[-1, 1]$ printr-un polinom de minimax de gradul 1.

Deoarece putem dezvolta funcția $\cos y$ în serie de puteri

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots + (-1)^n \frac{x^{2n}}{(2n)!} + \dots, x \in \mathbb{R}$$

avem folosind tabela care echivalează puterile lui x în raport cu polinoamele lui Cebîșev

$$\cos x \simeq T_0 - \frac{1}{2!} \left[\frac{1}{2} (T_0 + T_2) \right] + \frac{1}{4!} \left[\frac{1}{8} (3T_0 + 4T_2 + T_4) \right]$$

Neglijînd în continuare pe T_4 , care contribuie în mică măsură la formarea sumei, avem :

$$\cos x \simeq \frac{147}{192} T_0 - \frac{44}{192} T_2.$$

Înlocuind polinoamele Cebîșev T_0 și T_2 prin polinoamele în x corespunzătoare, obținem expresia aproximativă a polinomului de minimax de gradul doi care aproximează funcția $\cos x$:

$$\cos x \simeq \frac{191}{192} - \frac{88}{192} x^2.$$

În punctul $x = 0$ de exemplu, se obține o valoare aproximativă folosind polinomul astfel calculat egală cu 0,99479, față de valoarea exactă 1, deci o eroare egală cu 0,00521.

Aplicația 3

Să se aproximeze funcția $F(y) = y^2$ cu un polinom de minimax pe intervalul $[0, 1]$. Se folosește mai întîi transformarea

$$x = 2y - 1, \quad y \in [0, 1], \quad x \in [-1, 1]$$

pentru a transforma funcția $F(y)$ definită pe intervalul $[0, 1]$ într-o nouă funcție $f(x) = (1/4)(x^2 + 2x + 1)$ definită pe intervalul $[-1, 1]$.

Folosind tabela care face legătura între monoamele x^n și expresia corespunzătoare scrisă în funcție de polinoamele Cebîșev, obținem :

$$f(x) = \frac{1}{4} \left[\frac{1}{2} (T_0 + T_2) + 2T_1 + T_0 \right] = \frac{3}{8} T_0 + \frac{4}{8} T_1 + \frac{1}{8} T_2$$

Neglijind, conform teoriei, ultimul termen, rezultă :

$$f(x) \simeq \frac{3}{8} T_0 + \frac{4}{8} T_1 = \frac{3}{8} + \frac{4}{8} x.$$

Revenind la intervalul $[0, 1]$ se obține o aproximare a polinomului de minimax de forma :

$$F(y) \simeq y - 1/8.$$

9.2. Cazul discret

Fie funcția $f(x)$ dată printr-o tabelă de valori calculate în trei puncte diferite

x_k	x_1	x_2	x_3
f_k	f_1	f_2	f_3

Se poate arăta că există o dreaptă $y = mx + n$ care aproximează funcția dată astfel că în cele trei puncte funcția abatere ia valori egale ce alternează ca semn.

$$\Delta(x_1) = y_1 - f_1 = h$$

$$\Delta(x_2) = y_2 - f_2 = -h$$

$$\Delta(x_3) = y_3 - f_3 = h$$

Într-adevăr, deoarece în punctele x_k , ordonatele drepteii sint $y_k = mx_k + n$, se verifică egalitatea :

$$(x_3 - x_2)y_1 - (x_3 - x_1)y_2 + (x_2 - x_1)y_3 = 0.$$

În relația de mai sus să considerăm :

$$y_1 = f_1 + h, y_2 = f_2 - h, y_3 = f_3 + h.$$

Cu alte cuvinte, se alege dreapta ce trece prin punctele :

$$(x_1, f_1 + h), (x_2, f_2 - h), (x_3, f_3 + h),$$

deci are proprietatea menționată mai sus referitoare la funcția abatere. Rezultă :

$$(x_3 - x_2)(f_1 + h) - (x_3 - x_1)(f_2 - h) + (x_2 - x_1)(f_3 + h) = 0.$$

Se obține :

$$h = - \frac{(x_3 - x_1)f_1 - (x_3 - x_1)f_2 + (x_2 - x_1)f_3}{2(x_3 - x_1)}. \quad (c)$$

De fapt pentru construcția drepte sînt necesare doar două puncte, dar se verifică ușor că și cel de-al treilea punct este situat pe aceeași dreaptă. Dreapta astfel calculată poartă numele de *dreapta lui Cebîșev*. Se verifică că dreapta lui Cebîșev este unică, în plus este identică cu dreapta de minimax. Aceasta înseamnă că dreapta aproximează funcția dată în sensul lui Cebîșev, deci nu există o altă dreaptă care să realizeze o valoare maximă a abaterii mai mică, alternînd ca semn în cele trei puncte. În cazul în care funcția este dată printr-o tabelă în mai mult de trei puncte, pentru calculul drepte de minimax se recomandă o metodă iterativă. Aceeași metodă se folosește și în cazul căutării polinomului de minimax de grad superior. Se ilustrează metoda de calcul iterativ pe următorul exemplu :

Aplicația 4

Se dă funcția $f(x) = |x|$ printr-o tabelă de valori calculată în 5 puncte :

x_k	-2	-1	0	1	2
f_k	2	1	0	1	2

Se cere să se determine polinomul de minimax de gradul doi care aproximează această funcție folosind metoda iterativă.

Pasul 1

Metoda începe prin alegerea unui set inițial de patru puncte dintre cele date, de exemplu $-2, -1, 0, 1$.

Pasul 2

Se calculează în punctele selectate polinomul de gradul doi $p_2(x) = a + bx + cx^2$, conform teoriei lui Cebîșev, deci astfel ca abaterile $\Delta(x_k) = p_2(x_k) - f_k$ să aibă o valoare maximă egală (fie aceasta k) alternînd ca semn. Avem :

$$a - 2b + 4c - 2 = h$$

$$a - b + c - 1 = -h$$

$$a - 0 = h$$

$$a + b + c - 1 = -h$$

Soluția sistemului de ecuații va fi $a = 1/4$, $b = 0$, $c = 1/2$, $h = 1/4$.

Deci polinomul $p_2(x) = 1/4 + x^2/2$.

Pasul 3

Se calculează abaterile în toate cele cinci puncte, fie acestea m_k . Se notează prin M valoarea maximă absolută a acestora. Dacă $|h| = M$ procedura se termină, polinomul de aproximare fiind acela calculat la pasul 2. Este cazul exemplului de față, în care cele cinci abateri calculate sînt: $1/4$, $-1/4$, $1/4$, $-1/4$, $1/4$, deci în care valoarea maximă absolută este egală cu valoarea h calculată și anume $1/4$. Dacă acest fapt nu ar fi avut loc, algoritmul ar fi trebuit continuat în felul următor:

Pasul 4

Se alege un nou set de patru puncte, deci se modifică setul inițial în felul următor: se adaugă un nou punct la setul inițial și anume acela în care eroarea calculată este în valoare absolută, maximă. Apoi se renunță la un punct din setul inițial și anume astfel încît noul set de patru puncte să prezinte erori avînd semne alternante.

Sînt reluate pașurile 2 și 3. Se poate demonstra convergența procedurii descrise mai sus către polinomul de min-max. În cazul exemplului numeric prezent, parabola de min-max calculată aproximează funcția modul dată tabelar conform fig. 9.3.

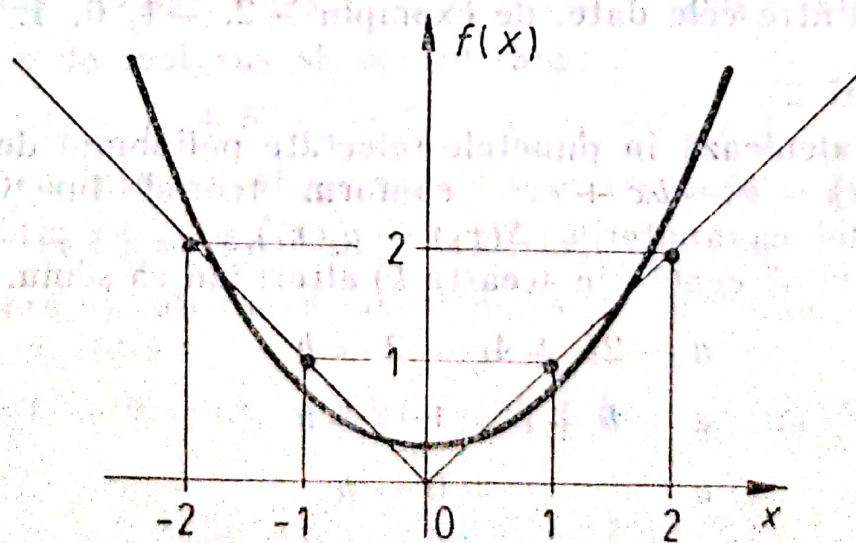


Fig. 9.3

10. SISTEME LINIARE SUPRADETERMINATE

10.1. Introducere

Sistemele *supradeterminate* sînt sisteme liniare avînd mai multe ecuații decît necunoscute. Apar de obicei în cazul unui număr prea mare de măsurători experimentale nu suficient de exacte.

Scris sub formă matricială, un sistem liniar supradeterminat are forma $Ax = b$, unde matricea sistemului A are mai multe linii decît coloane. Deoarece sistemul conține mai multe ecuații decît necunoscute, nu se poate găsi în general o soluție care să verifice exact toate ecuațiile sistemului. Din această cauză o soluție a unui sistem supradeterminat este un vector x , astfel ca vectorul rezidual: $r = Ax - b$ să fie într-un anumit sens „cît mai mic posibil“. În funcție de tipul aproximării se impun următoarele metode:

10.2. Soluția în sensul celor mai mici pătrate

Soluția sistemului supradeterminat $Ax = b$, este un vector x care face ca suma pătratelor componentelor vectorului rezidual r să fie minimă. În limbaj vectorial se impune

ca produsul $\mathbf{r}^T \cdot \mathbf{r}$ să fie minim. Fie de exemplu sistemul supra-determinat :

$$a_{11}x_1 + a_{12}x_2 = b_1$$

$$a_{21}x_1 + a_{22}x_2 = b_2$$

$$a_{31}x_1 + a_{32}x_2 = b_3$$

Pentru a pune în evidență componentele vectorului rezidual, rescriem sistemul sub forma :

$$a_{11}x_1 + a_{12}x_2 - b_1 = r_1$$

$$a_{21}x_1 + a_{22}x_2 - b_2 = r_2$$

$$a_{31}x_1 + a_{32}x_2 - b_3 = r_3$$

Rezultă că suma pătratelor componentelor vectorului rezidual este :

$$\begin{aligned} \Delta(x_1, x_2) = r_1^2 + r_2^2 + r_3^2 = & (a_{11}^2 + a_{21}^2 + a_{31}^2)x_1^2 + \\ & (a_{12}^2 + a_{22}^2 + a_{32}^2)x_2^2 + 2(a_{11}a_{12} + a_{21}a_{22} + a_{31}a_{32})x_1x_2 - \\ & - 2(a_{11}b_1 + a_{21}b_2 + a_{31}b_3)x_1 - 2(a_{12}b_1 + a_{22}b_2 + a_{32}b_3)x_2 \\ & + (b_1^2 + b_2^2 + b_3^2). \end{aligned}$$

Considerînd că expresia de mai sus este funcție de variabilele x_1 și x_2 , se impune în spiritul metodei celor mai mici pătrate ca derivatele parțiale ale expresiei de mai sus în raport cu variabilele x_1 și x_2 să se anuleze,

$$\frac{\partial \Delta}{\partial x_1} = 2 \langle \mathbf{a}_1, \mathbf{a}_1 \rangle x_1 + 2 \langle \mathbf{a}_1, \mathbf{a}_2 \rangle x_2 - 2 \langle \mathbf{a}_1, \mathbf{b} \rangle = 0;$$

$$\frac{\partial \Delta}{\partial x_2} = 2 \langle \mathbf{a}_2, \mathbf{a}_1 \rangle x_1 + 2 \langle \mathbf{a}_2, \mathbf{a}_2 \rangle x_2 - 2 \langle \mathbf{a}_2, \mathbf{b} \rangle = 0.$$

Se obține sistemul ce permite determinarea componentelor vectorului soluție $\mathbf{x}$.

$$\langle \mathbf{a}_1, \mathbf{a}_1 \rangle x_1 + \langle \mathbf{a}_1, \mathbf{a}_2 \rangle x_2 = \langle \mathbf{a}_1, \mathbf{b} \rangle$$

$$\langle \mathbf{a}_2, \mathbf{a}_1 \rangle x_1 + \langle \mathbf{a}_2, \mathbf{a}_2 \rangle x_2 = \langle \mathbf{a}_2, \mathbf{b} \rangle$$

Coeficienții sistemului de mai sus sînt produse scalare ale vectorilor (transpuși în prealabil) formați pe baza coloanelor matricei A sau coloana termenilor liberi. De exemplu, dacă

$$\mathbf{a}_1 = (a_{11}, a_{21}, a_{31}) \text{ și } \mathbf{a}_2 = (a_{12}, a_{22}, a_{32})$$

rezultă :

$$\langle \mathbf{a}_1, \mathbf{a}_2 \rangle = a_{11}a_{12} + a_{21}a_{22} + a_{31}a_{32}.$$

Metoda de mai sus se poate extinde și pentru un sistem de m ecuații avînd n necunoscute :

$$a_{11}x_1 + \dots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + \dots + a_{2n}x_n = b_2$$

$$a_{m1}x_1 + \dots + a_{mn}x_n = b_n$$

Procedînd în modul indicat, se ajunge la sistemul ce determină componentele vectorului soluție.

$$\begin{aligned} \langle \mathbf{a}_1, \mathbf{a}_1 \rangle x_1 + \langle \mathbf{a}_1, \mathbf{a}_2 \rangle x_2 + \dots + \langle \mathbf{a}_1, \mathbf{a}_n \rangle x_n &= \\ &= \langle \mathbf{a}_1, \mathbf{b} \rangle \end{aligned}$$

$$\begin{aligned} \langle \mathbf{a}_2, \mathbf{a}_1 \rangle x_1 + \langle \mathbf{a}_2, \mathbf{a}_2 \rangle x_2 + \dots + \langle \mathbf{a}_2, \mathbf{a}_n \rangle x_n &= \\ &= \langle \mathbf{a}_2, \mathbf{b} \rangle \end{aligned}$$

$$\begin{aligned} \langle \mathbf{a}_n, \mathbf{a}_1 \rangle x_1 + \langle \mathbf{a}_n, \mathbf{a}_2 \rangle x_2 + \dots + \langle \mathbf{a}_n, \mathbf{a}_n \rangle x_n &= \\ &= \langle \mathbf{a}_n, \mathbf{b} \rangle \end{aligned}$$

Aplicația 1

Să se determine soluția în sensul celor mai mici pătrate pentru sistemul

$$x_1 - 2x_2 = 1$$

$$x_1 + x_2 = 6$$

$$2x_1 - x_2 = 4$$

Se pun în evidență matricile

$$A = \begin{pmatrix} 1 & -2 \\ 1 & 1 \\ 2 & -1 \end{pmatrix} \text{ și } \mathbf{b} = \begin{pmatrix} 1 \\ 6 \\ 4 \end{pmatrix}$$

Sistemul care determină componentele vectorului soluție este de forma :

$$(1 + 1 + 4)x_1 + (-2 + 1 - 2)x_2 = 1 + 6 + 8$$

$$(-2 + 1 - 2)x_1 + (4 + 1 + 1)x_2 = -2 + 6 - 4$$

Rezultă :

$$6x_1 - 3x_2 = 15$$

$$-3x_1 + 6x_2 = 0$$

Soluția acestui sistem este dată de $x_1 = 30/9$ și $x_2 = 15/9$.

10.3. Soluție în sensul lui Cebîșev (soluție de minimax)

O soluție în sensul lui Cebîșev pentru sistemul supra-determinat $Ax = b$, este un vector x , astfel ca pentru vectorul rezidual $r = (r_1, r_2, \dots, r_m)$ să avem : $\max(|r_1|, |r_2|, \dots, |r_m|)$ să fie minimă.

Notînd valoarea maximă a expresiei de mai sus cu M , este evident că trebuie să avem :

$$|r_1| \leq M, \text{ deci } r_1 \leq M \text{ și } -r_1 \leq M$$

$$|r_2| \leq M, \text{ deci } r_2 \leq M \text{ și } -r_2 \leq M$$

$$|r_m| \leq M, \text{ deci } r_m \leq M \text{ și } -r_m \leq M$$

În cazul sistemului de 3 ecuații liniare cu două necunoscute componentele vectorului rezidual sînt date de sistemul :

$$a_{11}x_1 + a_{12}x_2 - b_1 = r_1$$

$$a_{21}x_1 + a_{22}x_2 - b_2 = r_2$$

$$a_{31}x_1 + a_{32}x_2 - b_3 = r_3$$

Atunci cele de mai sus conduc la următoarele 6 inegalități :

$$a_{11}x_1 + a_{12}x_2 - b_1 \leq M, \quad -a_{11}x_1 - a_{12}x_2 + b_1 \leq M$$

$$a_{21}x_1 + a_{22}x_2 - b_2 \leq M, \quad -a_{21}x_1 - a_{22}x_2 + b_2 \leq M$$

$$a_{31}x_1 + a_{32}x_2 - b_3 \leq M, \quad -a_{31}x_1 - a_{32}x_2 + b_3 \leq M$$

Se poate presupune ca $x_1, x_2 \geq 0$. În caz contrar se face o translație $z_i = z_i + k$ astfel încât să obținem primele două necunoscute pozitive. Ținând seama că se caută x_1, x_2 , astfel ca vectorul rezidual r să fie minim în sensul lui Cebîșev, este evident că determinarea necunoscutelor x_1, x_2 revine la rezolvarea unei probleme de programare liniară. Pentru o formulare mai simplă se împart inegalitățile de mai sus cu M , notînd :

$$y_1 = x_1/M, y_2 = x_2/M, y_3 = 1/M.$$

Atunci problema de programare liniară revine de a maximiza pe y_3 sau ceea ce este același lucru, de a face funcția obiectiv $F = 1/y_3$ minimă, astfel ca să fie verificat sistemul de condiții :

$$a_{11}y_1 + a_{12}y_2 - b_1y_3 \leq 1, \quad -a_{11}y_1 - a_{12}y_2 + b_1y_3 \leq 1$$

$$a_{21}y_1 + a_{22}y_2 - b_2y_3 \leq 1, \quad -a_{21}y_1 - a_{22}y_2 + b_2y_3 \leq 1$$

$$a_{31}y_1 + a_{32}y_2 - b_3y_3 \leq 1, \quad -a_{31}y_1 - a_{32}y_2 + b_3y_3 \leq 1$$

Condițiile pot fi scrise pornind direct de la sistemul de ecuații, fapt ce permite dezvoltarea metodei și pentru un sistem oarecare de m ecuații cu $n < m$ necunoscute. Comparînd cele două metode pentru căutarea soluției, este de remarcă că adesea metoda celor mai mici pătrate este mai ușor de utilizat.

Se recomandă să se utilizeze mai întîi această metodă și apoi dacă este necesar să se aplice metoda lui Cebîșev pentru a reduce în continuare vectorul rezidual.

CAPITOLUL 11

11. PROGRAMARE LINIARĂ

11.1 Introducere

O problemă de programare liniară cere să se determine valoarea maximă (sau minimă) a unei funcții liniare (funcția obiectiv):

$$f = c_1x_1 + \dots + c_nx_n$$

astfel ca relațiile de condiții (restricții) să fie verificate.

$$a_{i1}x_1 + \dots + a_{in}x_n \leq b_i, \quad i = 1, 2, \dots, m,$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n.$$

Cu notațiile matriciale:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}, \quad \mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix}$$

problema de programare liniară poate fi rescrisă astfel:

$$f(\mathbf{x}) = \mathbf{c}^T \mathbf{x} = \text{maxim (sau minim)}$$

$$A\mathbf{x} \leq \mathbf{b}$$

$$\mathbf{x} \geq 0$$

Aplicația 1

Se cere să se determine valoarea maximă a funcției obiectiv

$$f = x_1 + 4x_2$$

astfel încât să fie verificate restricțiile :

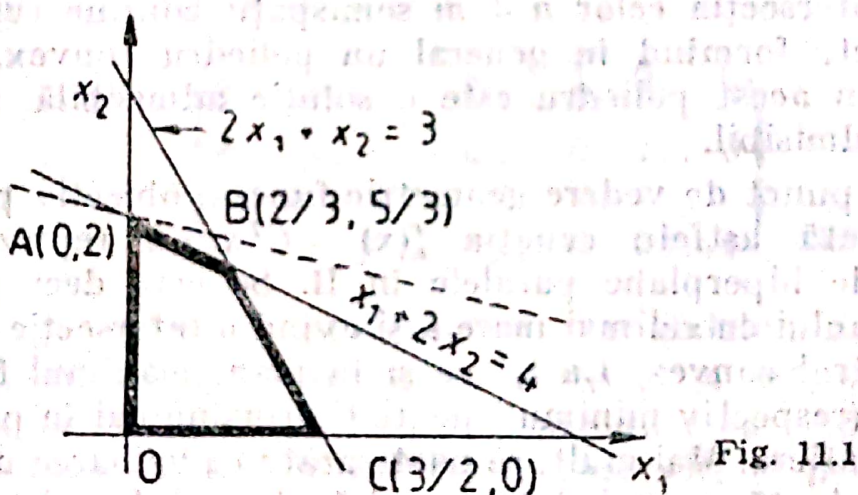
$$x_1 + 2x_2 \leq 4$$

$$2x_1 + x_2 \leq 3$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

Fiecare relație de condiție din setul de restricții reprezintă din punct de vedere geometric o dreaptă dacă se ia în considerare semnul egal. Reprezentând grafic aceste drepte, se obține poligonul convex $OABC$ (fig. 11.1).



Ținând seama că o dreaptă împarte planul în 2 semiplane pentru care expresia dreptei are semne contrarii, rezultă că fiecare punct din interiorul poligonului este o soluție posibilă a problemei, mulțimea acestor puncte formînd poligonul soluțiilor. Trebuie deci să alegem din această mulțime infinită de soluții posibile, pe acelea care fac funcția obiectiv maximă. Pentru aceasta să observăm că dacă scriem funcția obiectiv astfel :

$$x_1 + 4x_2 - f = 0$$

considerînd c  f reprezint  un parametru, atunci rela ia de mai sus reprezint  un fascicol de drepte paralele cu ordonatele la origin  propor ionale cu f . Evident func ia f va avea valoarea maxim  o dat  cu ordonata la origin  a dreptei trasat  punctat. Se ob ine :

$$f_{max} = x_1 + 4x_2 \Big|_{\substack{x_1=0 \\ x_2=2}} = 8$$

Se observ  c  solu ia optim  are loc pentru coordonatele unui v rf al poligonului solu iilor. Pe de alt  parte problema are solu ia unic , deoarece fascicolul de drepte $x_1 + 4x_2 - f = 0$ nu este paralel cu nici una din laturile poligonului.

Acceast  metod  denumit  *metoda geometric *, poate fi folosit  numai c nd se lucreaz  cu dou  sau trei variabile. Generalizarea se face astfel : un sistem general de $n + m$ restric ii define te $n + m$ semispa ii  nchise, punctul x fiind interpretat ca un punct sau un vector  ntr-un spa iu n dimensional $\mathbf{R}^n$. Dac  cele $n + m$ restric ii s nt consistente, atunci intersec ia celor $n + m$ semispa ii con ine cel pu in un punct, form nd  n general un poliedru convex. Orice punct din acest poliedru este o solu ie admisibil  sau un vector admisibil.

Din punct de vedere geometric func ia obiectiv poate fi interpretat  astfel : ecua ia $f(x) = C^T x = k$ reprezint  o familie de hiperplane paralele  n $\mathbf{R}$. Se cere deci g sirea hiperplanului cu cel mai mare k  i av nd o intersec ie nevid  cu poliedrul convex. La fel ca  i  n plan, maximul func iei obiectiv (respectiv minimul) poate fi atins numai  n punctele de pe frontier . Mai mult, se poate ar ta c  valoarea maxim  poate fi luat  numai  ntr-un v rf al poliedrului convex. Fie x_k , $k = 1, 2, \dots, r$ v rfurile poliedrului  n num r de cel mult C_{n+m}^n . Atunci oricare ar fi punctul x din interiorul poliedrului convex, avem :

$$x = \sum_{k=1}^r \alpha_k x_k, \text{ cu } \alpha_k \geq 0 \text{  i } \sum_{k=1}^r \alpha_k = 1.$$

 n plus :

$$f(x) = c^T x = c^T \sum_{k=1}^r \alpha_k x_k = \sum_{k=1}^r \alpha_k c^T x_k = \sum_{k=1}^r \alpha_k f(x_k)$$

Dar printre cele r valori ale funcției f în vîrfurile poliedrului există o valoare maximă $f(x_M)$. Deci :

$$f(x) = \sum_{k=1}^r \alpha_k f(x_k) \leq \sum_{k=1}^r \alpha_k f(x_M) = f(x_M).$$

Deci problema de optimizare revine la calculul vîrfurilor poliedrului convex, determinarea funcției obiectiv pentru fiecare vîrf și selectarea vîrfului pentru care f ia valoarea optimă. Pentru aceasta se introduce mai întii m variabile auxiliare nenegative $x_{n+1}, \dots, x_{n+m}$ care transformă sistemul de restricții în egalități. Se spune că problema de programare liniară a fost adusă la forma normală :

$$f(x) = c^T x = \text{minimum (sau maximum)}$$

$$Ax = b$$

$$x \geq 0$$

unde matricea A este de forma :

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} & 1 & 0 \dots 0 \\ a_{21} & a_{22} & \dots & a_{2n} & 0 & 1 \dots 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} & 0 & 0 \dots 1 \end{pmatrix}$$

ultimele m coloane corespunzînd variabilelor auxiliare.

Se poate presupune că rangul matricei A este m pentru a exista o soluție admisibilă. Introducerea variabilelor auxiliare face posibilă identificarea unui vîrf al poliedrului convex, sau, așa cum se mai numește, a unei soluții admisibile de bază. Într-adevăr, inegalitatea $Ax \leq b$ corespunzînd acum unei situații în care variabilele auxiliare sînt zero, un vîrf (soluție admisibilă de bază) corespunde unui punct x în care cel puțin n variabile $x_1, \dots, x_{n+m}$ sînt zero. Ori, altfel spus, un punct x reprezintă o soluție admisibilă de bază dacă cel mult m componente sînt nenegative. Printre aceste soluții admisibile de bază, în număr de cel mult C_{m+n}^n se găsește cea valoare în care funcția obiectiv ia valoare maximă. Acest punct nu este în mod necesar unic determinat.

Să presupunem în continuare că primele m coloane ale matricei A sînt linear independente și să notăm cu A_1 matricea nesingulară de ordinul $m \times m$ ce conține aceste coloane, iar cu A_2 matricea de ordinul $m \times n$ ce conține restul coloanelor din A . Deci, $A = (A_1, A_2)$. În mod analog se separă vectorii $c = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}$, $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$, unde c_1 și x_1 se compun din primele m componente. Ecuația $Ax = b$ devine astfel $A_1x_1 + A_2x_2 = b$. Rezolvînd ultima ecuație în raport cu x_1 , se obține :

$$x_1 = A_1^{-1}b + A_1^{-1}A_2(-x_2).$$

Pentru vectorul x_1 funcția obiectiv devine :

$$\begin{aligned} f(x) &= (c_1^T, c_2^T) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = c_1^T x_1 + c_2^T x_2 = \\ &= c_1^T A_1^{-1}b + [c_1^T A_1^{-1}A_2 - c_2^T](-x_2). \end{aligned}$$

Dacă se presupune $A_1^{-1}b > 0$ și $x_2 = 0$, atunci se obține o soluție admisibilă de bază, deoarece în acest caz punctul x are cel mult m componente nenegative.

11.2. Metoda simplex

Rezultatele din paragraful precedent se grupează de obicei într-o tabelă simplex, metoda care utilizează aceste tabele pentru căutarea soluției numindu-se *metoda simplex*.

Tabel simplex	$A_1^{-1}b$	$A_1^{-1}A_2$
	$c_1^T A_1^{-1}b$	$c_1^T A_1^{-1}A_2 - c_2^T$

Se observă că un tabel simplex dă în prima coloană atât pe $A_1^{-1}b$, adică vectorul ce constituie o soluție admisibilă de bază, cît și pe $c_1^T A_1^{-1}b$, adică valoarea funcției obiectiv în acest punct. Metoda simplex se desfășoară în următoarele etape :

- se determină mai întîi o soluție inițială de bază ;

— se înlocuiește iterativ baza inițială cu alte baze astfel ca pentru fiecare nouă bază, funcția obiectiv să se majoreze (sau să se reducă în caz de minim). După un număr de iterații se ajunge la situația că funcția obiectiv nu se mai poate majora (sau reduce). Ultima bază calculată corespunde unei soluții optime.

Aplicația 2

Se cere să se illustreze metoda simplex pentru problema de programare liniară din aplicația 1.

Folosind variabilele auxiliare $x_3 > 0$ și $x_4 > 0$, problema de programare liniară este adusă la forma normală

$$f = 0 \cdot x_3 + 0 \cdot x_4 + x_1 + 4x_2.$$

Sistemul de restricții devine:

$$1 \cdot x_3 + 0 \cdot x_4 + x_1 + 2x_2 = 4$$

$$0 \cdot x_3 + 1 \cdot x_4 + 2x_1 + x_2 = 3.$$

Se obțin:

$$A = \begin{pmatrix} 1 & 0 & 1 & 2 \\ 0 & 1 & 2 & 1 \end{pmatrix}, A_1 = A_1^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, A_2 = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$$

$$e_1^T = (0, 0), e_2^T = (1, 4), x_1^T = (x_3, x_4), x_2^T = (x_1, x_2)$$

Dacă rescriem sistemul de restricții astfel:

$$x_3 = 4 - x_1 - 2x_2$$

$$x_4 = 3 - 2x_1 - x_2$$

se observă că pentru $x_1 = x_2 = 0$ se obține $x_3 = 4$, $x_4 = 3$ care este o soluție admisibilă de bază, ce poate fi luată drept bază inițială. Se poate scrie deci primul tabel simplex:

	4	1	2
S_1	3	2	1
	0	-1	-4

Se face în continuare o schimbare de bază. Deoarece coeficienții lui x_1 și x_2 sînt pozitivi în expresia funcției obiectiv,

aceasta va crește dacă se consideră o altă bază care să conțină pe x_1 sau x_2 . Deoarece coeficientul lui x_2 este mai mare decât coeficientul lui x_1 , se introduce x_2 în noua bază. Se înlocuiește deci vechea bază (x_3, x_4) cu noua bază (x_2, x_4) .

Rezolvând sistemul de restricții în raport cu x_1 și x_3 , variabile din afara bazei, se obține:

$$x_2 = 2 - \frac{1}{2}x_1 - \frac{1}{2}x_3$$

$$x_4 = 3 - 2x_1 - x_3$$

Pentru $x_1 = x_3 = 0$, se obține $x_2 = 2$, $x_4 = 3$ care este o nouă soluție admisibilă de bază. Se obțin:

$$A_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = A_1^{-1}, \quad A_2 = \begin{pmatrix} 1/2 & 1/2 \\ 2 & 1 \end{pmatrix}$$

$$c_1^T = (4, 0), \quad c_2^T = (1, 0), \quad x_1^T = (x_2, x_4), \quad x_2^T = (x_1, x_3)$$

Se poate scrie un nou tabel simplex

	2	1/2	1/2
S_2	4	2	1
	8	1	2

Se obține deci valoarea maximă pentru funcția obiectiv egală cu 8, adică regăsim valoarea calculată și prin metoda geometrică. Ultima bază calculată și anume $(2, 4)$ corespunde unei soluții optime a problemei de programare liniară, deoarece toate componentele vectorului $c_1^T A_1^{-1} A_2 - c_2^T$ care se găsește în căsuța situată în poziția dreapta jos sînt nenegative, fapt ce semnaleză că algoritmul simplex a ajuns la etapa finală.

Dacă am fi introdus în bază pe x_1 în locul lui x_2 , de exemplu, noua bază (x_1, x_4) nu ar fi fost acceptabilă, deoarece ar fi condus la soluții de bază admisibile cu elemente negative, fapt ce ar fi contrazis ipoteza de lucru (toate argumentele $x_k \geq 0$).

CAPITOLUL 12

12. DERIVAREA NUMERICĂ

12.1. Introducere

Deoarece derivarea este în general o operație ușor de efectuat, se recurge la derivare numerică în situația în care funcția nu este cunoscută analitic, ci dată printr-o tabelă de valori.

$$\begin{array}{c|c} x_k & x_0 \ x_1 \dots x_n \\ \hline f(x_k) & f(x_0)f(x_1) \dots f(x_n) \end{array}$$

Cel mai simplu mod de a realiza derivarea numerică a funcției de mai sus este de a o aproxima mai întâi printr-un polinom de aproximare, urmînd să derivăm apoi acest polinom, adică :

$$f(x) \simeq p_n(x)$$

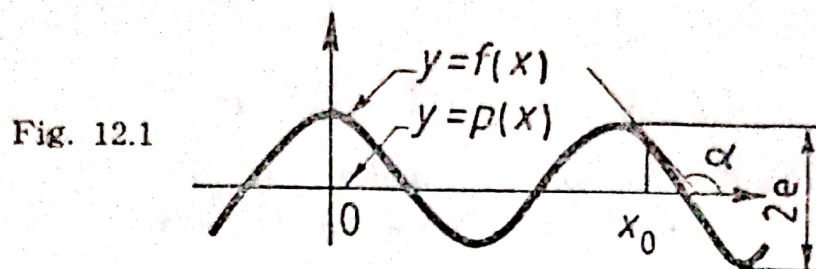
$$\frac{df}{dx} \simeq \frac{dp_n(x)}{dx}$$

$$\vdots$$

$$\frac{d^n f}{dx^n} \simeq \frac{d^n p_n(x)}{dx^n}$$

Se face însă observația că derivarea numerică tinde să crească micile erori în aproximarea funcției. În limbaj geometric, două curbe pot fi destul de „aproape” una de alta și totuși mărimea pantei tangentelor duse la cele două curbe

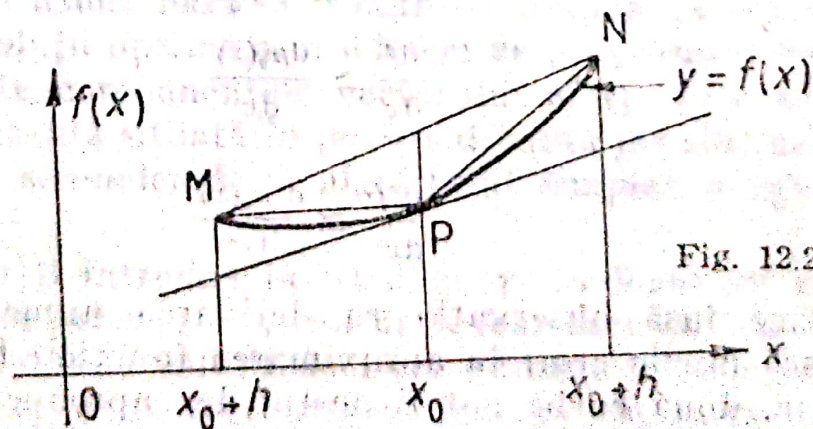
într-un punct comun (derivata întâi) să difere considerabil. De exemplu, (fig. 12.1) funcția $f(x) = e \cos \frac{x}{e^2}$ este bine aproximată de polinomul $p(x) = 0$ cu care coincide în punctele $x_k = (2k + 1)\pi e^2/2$; eroarea maximă nu depășește valoarea e , totuși, deoarece $f'(x) = -\frac{1}{e} \sin \frac{x}{e^2}$ și $p'(x) = 0$, diferența între derivate este foarte mare.



Dacă calculul numeric al derivatei întâi este o operație riscantă, derivările succesive ulterioare pot crește nivelul erorilor pînă la un nivel inadmisibil. Pornind de la definiția derivatei și renunțînd la operația de trecere la limită, obținem două formule simple pentru calculul aproximativ al derivatei întâi :

$$f'(x_0) \simeq \frac{f(x_0 + h) - f(x_0)}{h}, \quad f'(x) \simeq \frac{f(x_0) - f(x_0 - h)}{h}$$

Altă formulă simplă folosită adesea pentru calculul aproximativ al lui $f'(x_0)$ se bazează pe fig. 12.2, observînd că panta tangentei dusă la curbă în punctul x_0 este mai bine aproximată prin panta coardei MN decît prin panta coardelor PN și PM ce servesc ca suport geometric pentru formulele precedente.



Avem :

$$f'(x_0) \simeq \frac{f(x_0 + h) - f(x_0 - h)}{2h}$$

Pentru a evalua eroarea care se comite folosind în calcule formula aproximativă de derivare de mai sus, se dezvoltă în serie Taylor funcția $f(x)$ în vecinătatea punctului x_0 .

$$f(x_0 + h) = f(x_0) + hf'(x_0) + \frac{h^2}{2!} f''(x_0) + \frac{h^3}{6} f'''(\xi_1),$$

$$\xi_1 \in (x_0, x_0 + h)$$

$$f(x_0 - h) = f(x_0) - hf'(x_0) + \frac{h^2}{2} f''(x_0) - \frac{h^3}{6} f'''(\xi_2),$$

$$\xi_2 \in (x_0 - h, x_0)$$

Scăzând termen cu termen cele două formule se obține :

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0 - h)}{2h} - \frac{h^2}{6} \frac{f'''(\xi_1) + f'''(\xi_2)}{2}$$

Fie α cuprins între ξ_1 și ξ_2 , astfel încît :

$$\frac{f'''(\xi_1) + f'''(\xi_2)}{2} = f'''(\alpha)$$

Să presupunem că $f'''(x)$ este mărginită pe $(x_0 - h, x_0 + h)$.
Atunci :

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0 - h)}{2h} - \frac{h^2}{6} f'''(\alpha),$$

$$x_0 - h < \alpha < x_0 + h.$$

Rezultă că eroarea de trunchiere este de forma :

$$\epsilon_T = \frac{h^2}{6} f'''(\alpha), \quad x_0 - h < \alpha < x_0 + h$$

Din expresia erorii obținută, rezultă că micșorînd pe h , se micșorează și eroarea. În realitate nu se întîmplă întotdeauna astfel, deoarece micșorînd pe h , numerele $f(x_0 + h)$ și $f(x_0 - h)$ pot deveni aproximativ egale și în acest caz conform celor spuse în capitolul 1 eroarea de rotunjire crește foarte mult.

12.2. Derivare numerică folosind polinoamele de aproximare

Pentru a obține formule de calcul mai precise trebuie să folosim polinoamele de interpolare. De exemplu, dacă aproximăm funcția dată prin polinomul lui Newton de interpolare cu diferențe finite la dreapta, obținem :

$$f(x) \simeq y_0 + \frac{\Delta y_0}{1!h} (x - x_0) + \frac{\Delta^2 y_0}{2!h^2} (x - x_0)(x - x_1) + \\ + \dots + \frac{\Delta^n y_0}{n!h^n} (x - x_0)(x - x_1) \dots (x - x_{n-1})$$

Efectuând înmulțirile între paranteze, avem :

$$f(x) \simeq y_0 + \frac{\Delta y_0}{1!h} (x - x_0) + \frac{\Delta^2 y_0}{2!h^2} [x^2 - (x_0 + x_1)x + x_0x_1] + \\ + \frac{\Delta^3 y_0}{3!h^3} [x^3 - (x_0 + x_1 + x_2)x^2 + (x_0x_1 + x_1x_2 + x_0x_2)x - x_0x_1x_2] + \dots$$

Derivând succesiv expresia anterioară, obținem :

$$f'(x) \simeq \frac{\Delta y_0}{1!h} + \frac{\Delta^2 y_0}{2!h^2} [2x - (x_0 + x_1)] + \\ + \frac{\Delta^3 y_0}{3!h^3} [3x^2 - 2x(x_0 + x_1 + x_2) + (x_0x_1 + x_1x_2 + \\ + x_0x_2)] + \dots$$

$$f''(x) \simeq \frac{\Delta^2 y_0}{h^2} + \frac{\Delta^3 y_0}{3!h^3} [6x - 2(x_0 + x_1 + x_2)] + \dots$$

Continuând în același fel, obținem formula pentru calculul aproximativ al derivatelor lui $f(x)$ de un ordin oarecare. Dacă se dorește să se calculeze valorile aproximative ale derivatelor lui $f(x)$ în unul din punctele $x_0, x_1, \dots, x_n$

formulele de mai sus se simplifică. De exemplu, pentru $x = x_0$, avem :

$$f'(x_0) \simeq \frac{1}{h} \left(\Delta y_0 - \frac{\Delta^2 y_0}{2} + \frac{\Delta^3 y_0}{3} - \frac{\Delta^4 y_0}{4} + \dots \right)$$

$$f''(x_0) \simeq \frac{1}{h^2} \left(\Delta^2 y_0 - \Delta^3 y_0 + \frac{11}{12} \Delta^4 y_0 + \dots \right)$$

Se pot calcula aproximativ expresiile erorilor de derivare. Astfel, deoarece :

$$\begin{aligned} \varepsilon_n(x) = f(x) - p_n(x) &\simeq \frac{\Delta^{n+1} y_0}{h^{n+1}(n+1)!} (x - x_0)(x - \\ &- x_1) \dots (x - x_n), \end{aligned}$$

$$\frac{d}{dx} [\varepsilon_n(x)] = f'(x) - p'_n(x) \simeq \frac{d}{dx} \left[\frac{\Delta^{n+1} y_0}{h^{n+1}(n+1)!} (x - x_0)(x - \right.$$

$$\left. - x_1) \dots (x - x_n) \right]$$

Notînd $t = (x - x_0)/h$, adică $x = x_0 + th$, rezultă :

$$x - x_0 = ht$$

$$x - x_1 = x - x_0 - h = ht - h = h(t - 1)$$

$$x - x_n = h(t - n)$$

Se obține :

$$\varepsilon_n(x_0 + th) \simeq \frac{\Delta^{n+1} y_0}{(n+1)!} t(t-1) \dots (t-n)$$

Deoarece :

$$\frac{d}{dx} [\varepsilon_n(x_0 + th)] = \frac{d\varepsilon_n(x_0 + th)}{dt} \cdot \frac{dt}{dx} = \frac{1}{h} \frac{d\varepsilon_n(x_0 + th)}{dt};$$

$$[t(t-1) \dots (t-n)]'_{t=0} = (-1)^n n!$$

În final rezultă expresia erorii de forma :

$$f'(x) - p'_n(x) = (-1)^n \frac{\Delta^{n+1} y_0}{(n+1)h}$$

Urmărind calculul aproximativ al derivatelor de ordin superior de mai sus se poate pune în evidență următoarea formulă simplă pentru calculul derivatei de ordinul n :

$$y^{(n)}(x_0) \simeq \frac{\Delta^n y_0}{h^n}$$

Se pot folosi și diferențe finite la stînga sau centrale pentru a pune în evidență formule de calcul aproximativ al derivatelor.

Astfel, deoarece

$$y_i'' \simeq \frac{\delta^2 y_i}{h^2}$$

și ținînd seama că

$$\delta^2 y_i = y_{i+1} - 2y_i + y_{i-1},$$

rezultă o formulă mult folosită în aproximarea prin metoda rețelei (vezi capitolul 17) a ecuațiilor cu derivate parțiale de ordinul doi :

$$y_i'' \simeq \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}$$

Rezultate mai bune se pot obține în derivarea numerică utilizînd în locul polinoamelor de interpolare polinoame de minimax sau cu abatere medie pătratică minimă. Astfel, pentru funcția dată tabelar

$$\begin{array}{c|cccccc} x_k & x_0 & x_1 & \dots & x_k & \dots & x_{n-1} & x_n \\ \hline y_k & y_0 & y_1 & \dots & y_k & \dots & y_{n-1} & y_n \end{array}$$

folosind un polinom de aproximare $p_2(x) = c_0 + c_1x + c_2x^2$ cu abatere medie pătratică minimă calculată pe baza a cinci puncte (x_i, y_i) unde $i = k-2, k-1, k, k+1, k+2$ (a se vedea aplicația 4, capitolul 8) se poate pune în evidență formula de derivare în punctele x_k :

$$y'(x_k) \simeq \frac{1}{10h} (-2y_{k-2} - y_{k-1} + y_{k+1} + 2y_{k+2})$$

$$k = 2, 3, \dots, n-2$$

Deoarece formula de mai sus implică utilizarea a două valori x situate la stînga și la dreapta lui x_k pentru punctele situate la extremitățile tabelului, se recomandă să se folosească formulele de calcul date mai jos obținute asemenea dar lucrînd numai cu seturi de patru puncte x_0, x_1, x_2, x_3 respectiv $x_{n-3}, x_{n-2}, x_{n-1}, x_n$.

$$y'(x_0) \simeq \frac{1}{20h} (-21y_0 + 13y_1 + 17y_2 - 9y_3)$$

$$y'(x_1) \simeq \frac{1}{20h} (-11y_0 + 3y_1 + 7y_2 + y_3)$$

$$\vdots$$

$$y'(x_{n-1}) \simeq \frac{1}{20h} (11y_n - 3y_{n-1} - 7y_{n-2} - y_{n-3})$$

$$y'(x_n) \simeq \frac{1}{20h} (21y_n - 13y_{n-1} - 17y_{n-2} + 9y_{n-3})$$

Se poate arăta că formulele de mai sus sînt mult mai precise decît formulele echivalente obținute prin derivarea polinoamelor de interpolare.

CAPITOLUL 13

13. INTEGRARE NUMERICĂ

13.1. Formule de cuadratură numerică

Există deseori în practică situații când o funcție ce urmează să fie integrată este dată tabelar sau când o primitivă a sa este fie prea complicată, fie nu poate fi exprimată cu ajutorul funcțiilor elementare ca de exemplu în cazul funcției e^{-x^2} . În aceste situații se recurge la calculul aproximativ al integralei. Una dintre cele mai simple modalități de calcul aproximativ al integralei $\int_a^b f(x) dx$ constă în înlocuirea funcției $f(x)$ integrabilă pe intervalul $[a, b]$ printr-un polinom de aproximare, apoi integrarea acestui polinom. Cu alte cuvinte :

$$\int_a^b f(x) dx \simeq \int_a^b p_n(x) dx$$

Polinomul de aproximare $p_n(x)$ poate fi un polinom de interpolare, dar și un polinom de minimax sau de aproximare cu abatere medie pătratică minimă. Spre deosebire de derivarea numerică ce tinde să crească micile erori în aproximarea funcției, integrarea numerică în aceeași măsură tinde să le atenueze. Astfel, dacă aproximarea polinomială este

rezonabil de bună, integrala $\int_a^b p_n(x) dx$ poate fi o aproximație excelentă pentru $\int_a^b f(x) dx$.

Să facem o diviziune a intervalului $[a, b]$ cu noduri echidistante $a = x_0, < x_1 < x_2 < \dots < x_n = b$. Folosind proprietatea de aditivitate a integralei față de interval, se poate scrie :

$$I = \int_a^b f(x) dx = \sum_{k=0}^{n-1} \int_{x_k}^{x_{k+1}} f(x) dx$$

Să aproximăm în continuare funcția $f(x)$ pe intervalul $[x_k, x_{k+1}]$ printr-un polinom Newton de interpolare de gradul unu.

$$f(x) \simeq y_k + \frac{\Delta y_k}{1! h} (x - x_k),$$

unde h reprezintă distanța dintre punctele x_k și x_{k+1} .
Avem :

$$I_k = \int_{x_k}^{x_{k+1}} f(x) dx \simeq \int_{x_k}^{x_{k+1}} \left[y_k + \frac{\Delta y_k}{1! h} (x - x_k) \right] dx$$

Să facem schimbarea de variabilă $(x - x_k)/h = t$
Se obține :

$$I_k \simeq h \int_0^1 (y_k + t \Delta y_k) dt = h \left(y_k + \frac{\Delta y_k}{2} \right)$$

Deoarece :

$$\Delta y_k = y_{k+1} - y_k, \text{ se obține } I_k \simeq \frac{h}{2} (y_k + y_{k+1})$$

Rezultă :

$$\int_a^b f(x) dx \simeq \frac{h}{2} \sum_{k=0}^{n-1} (y_k + y_{k+1})$$

Deci :

$$\int_a^b f(x) dx \simeq \frac{h}{2}(y_0 + 2y_1 + 2y_2 + \dots + 2y_{n-1} + y_n)$$

Formula astfel obținută poartă numele de *formula trapezelor*, deoarece (fig.13.1) se aproximează integrala reprezentată geometric prin aria cuprinsă între curba $y = f(x)$ și ordonatele ridicate în $x = a$ și $x = b$ prin suma ariilor trapezelor construite.

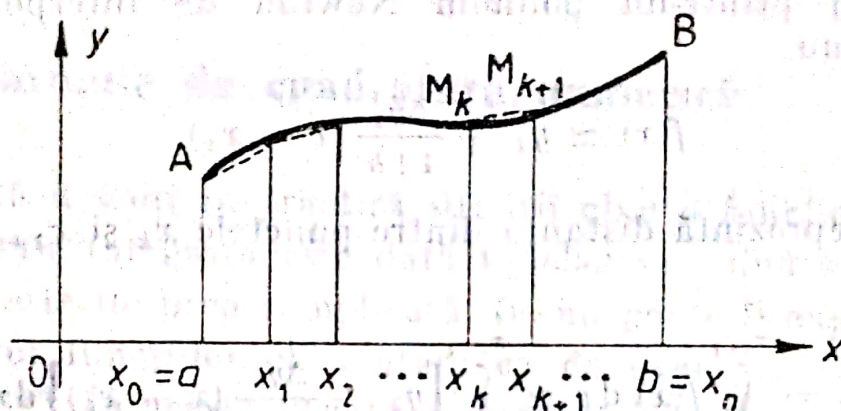


Fig. 13.1

Să calculăm eroarea pe care o implică utilizarea formulei trapezelor.

Fie $F(x)$ o primitivă pentru funcția integrant $f(x)$. Conform formulei Leibniz-Newton, avem pentru intervalul $[x_k, x_{k+1}]$:

$$\begin{aligned} \varepsilon_k(f) &= \int_{x_k}^{x_{k+1}} f(x) dx - \frac{h}{2} [f(x_k) + f(x_{k+1})] = \\ &= F(x_{k+1}) - F(x_k) - \frac{h}{2} [f(x_k) + f(x_{k+1})] \end{aligned}$$

În continuare se dau dezvoltările în serie Taylor a funcțiilor $F(x)$ și $f(x)$ în jurul punctului x_k folosind de asemenea faptul că $F(x)$ reprezintă o primitivă pentru $f(x)$:

$$f(x_{k+1}) = f(x_k) + \frac{h}{1!} f'(x_k) + \frac{h^2}{2!} f''(\xi_k)$$

$$F(x_{k+1}) = F(x_k) + \frac{h}{1!} f(x_k) + \frac{h^2}{2!} f'(x_k) + \frac{h^3}{3!} f''(\xi_k)$$

Introducînd relațiile de mai sus în expresia erorii ε_k , se obține:

$$\varepsilon_k(f) = -\frac{h^3}{12} f''(\xi_k)$$

Eroarea totală va fi:

$$\varepsilon_T(f) = \sum_{k=0}^{n-1} \varepsilon_k(f) = -\frac{h^3}{12} \sum_{k=0}^{n-1} f''(\xi_k) = -\frac{(b-a)^3}{12n^2} f''(\xi)$$

unde punctul $\xi \in (a, b)$ este astfel ales, încît:

$$f''(\xi) = \frac{1}{n} \sum_{k=0}^{n-1} f''(\xi_k)$$

Dacă se presupune că derivata a doua a funcției f admite o majorantă, adică există $M > 0$, astfel că $|f''(x)| \leq M$ oricare ar fi $x \in (a, b)$, rezultă:

$$|\varepsilon_T(f)| \leq \frac{M(b-a)^3}{12n^2}.$$

Formula trapezelor a fost obținută aproximînd funcția integrant cu un polinom Newton de interpolare de gradul unu. Dacă se folosește un polinom de interpolare de gradul doi procedînd ca mai sus și folosind o diviziune a lui $[a, b]$ într-un număr par de puncte, se poate pune în evidență

una din cele mai des folosite formule de integrare numerică numită *formula lui Simpson* :

$$\int_a^b f(x) dx \simeq \frac{h}{3} [y_0 + 4y_1 + 2y_2 + 4y_3 + \dots + 2y_{n-2} + 4y_{n-1} + y_n]$$

Eroarea pe care o implică aproximarea integralei cu formula lui Simpson este :

$$|\varepsilon_T(f)| \leq \frac{M(b-a)^5}{180n^4}$$

Comparînd expresiile erorilor pentru cele două formule de integrare numerică rezultă că formula lui Simpson dă pentru același număr de noduri o precizie mai bună decît formula trapezelor. Folosind polinoame de interpolare de grad superior, se pot pune în evidență și alte formule de integrare numerică, (numite și *formule de cuadratură numerică*) mai precise.

Aplicația 1

Să se estimeze efectul greșelilor în măsurarea datelor atunci cînd se aplică formula trapezelor.

Notînd prin Y_k valorile exacte și prin y_k valorile măsurate ale ordonatelor, avem :

$$Y_k - y_k = \varepsilon_k, \text{ deci } Y_k = y_k + \varepsilon_k,$$

unde ε_k reprezintă eroarea de măsurare pentru ordonata y_k .
Aplicînd formula trapezelor, obținem :

$$\begin{aligned} \frac{h}{2} (Y_0 + 2Y_1 + \dots + 2Y_{n-1} + Y_n) &= \frac{h}{2} (y_0 + 2y_1 + \\ &+ \dots + 2y_{n-1} + y_n) + \frac{h}{2} (\varepsilon_0 + 2\varepsilon_1 + \dots + 2\varepsilon_{n-1} + \varepsilon_n) \end{aligned}$$

Dacă

$$|\varepsilon_k| \leq M, \quad (\forall) k = 0, 1, 2, \dots, n$$

rezultă :

$$\varepsilon_T \leq \frac{h}{2} [M + 2(n-1)M + M] = (b-a)M.$$

13.2. Metoda lui Romberg și procedura de extrapolare a lui Richardson

Metoda lui Romberg își propune să îmbunătățească eficiența formulelor de cuadratură numerică aplicând repetat una din formulele de calcul, în asociere cu înjumătățirea în același timp a intervalului h (deci dublind numărul de puncte ale diviziunii). Într-adevăr, folosind de exemplu formula trapezelor, eroarea este proporțională cu h^2 . Înjumătățind pe h și reaplicând formula trapezelor, eroarea se diminuează de 4 ori. Să notăm cu $I_{k,1}$ valoarea calculată a integralei $\int_a^b f(x) dx$, folosind formula trapezelor cu $n = 2^k$. Lucrând doar cu punctele $a = x_0, x_1 = b$, avem: $h = (b - a)$, $n = 1$, $k = 0$, deci: $I_{0,1} = \frac{b-a}{2} [f(a) + f(b)]$.

În continuare, înjumătățim pe h introducând între a și b punctul $x_1 = (b - a)/2$, deci lucrăm cu 3 puncte $a = x_0, x_1, x_2 = b$. Avem $h = (b - a)/2$, $n = 2$, $k = 1$. Rezultă:

$$\begin{aligned} I_{1,1} &= \frac{b-a}{4} \left[f(a) + 2f\left(a + \frac{b-a}{2}\right) + f(b) \right] = \\ &= \frac{1}{2} \left[I_{0,1} + (b-a)f\left(a + \frac{b-a}{2}\right) \right]. \end{aligned}$$

Înjumătățind în continuare intervalul h , se obține:

$$\begin{aligned} I_{2,1} &= \frac{b-a}{8} \left[f(a) + f(b) + 2 \sum_{i=1}^3 f\left(a + \frac{b-a}{4}i\right) \right] = \\ &= \frac{1}{2} \left[I_{1,1} + \frac{b-a}{2} \sum_{i=1}^3 f\left(a + \frac{b-a}{4}i\right) \right]. \end{aligned}$$

Prin inducție se obține în final valoarea $I_{k,1}$:

$$I_{k,1} = \frac{1}{2} \left[I_{k-1,1} + \frac{b-a}{2^{k-1}} \sum_{i=1}^{2^{k-1}} f\left(a + \frac{b-a}{2^k}i\right) \right]$$

Eroarea ce corespunde lui $I_{k,1}$ este :

$$\varepsilon_{k,1} = - \frac{(b-a)^3}{12(2)^{2k}} f''(\xi), \quad \xi \in (a, b)$$

Presupunind că funcția $f(x)$ are o derivată de ordinul doi mărginită în intervalul (a, b) , relația de mai sus ne asigură că secvența : $I_{0,1}, I_{1,1}, \dots, I_{k,1}, \dots$ converge către valoarea exactă. Rapiditatea convergenței poate fi mărită folosind procedura de extrapolare a lui Richardson. Să presupunem că se folosește o formulă de cvadratură pentru care eroarea este proporțională cu h^n . Atunci două aplicații succesive ale acestei formule lucrând cu intervalele $2h$ și h conduc la erorile : $\varepsilon_1 \simeq K_1(2h)^n$, $\varepsilon_2 \simeq K_2(h)^n$. Presupunind factorii de proporționalitate K_1 și K_2 , egali, obținem :

$$\varepsilon_2 \simeq \frac{\varepsilon_1}{2^n}$$

Dacă valoarea exactă a integralei o notăm cu I , avem :

$$I = I_1 + \varepsilon_1 = I_2 + \varepsilon_2$$

Deci :

$$I_1 + 2^n \varepsilon_2 \simeq I_2 + \varepsilon_2$$

Rezultă :

$$\varepsilon_2 \simeq \frac{I_2 - I_1}{2^n - 1}$$

În final :

$$I \simeq I_2 + \frac{I_2 - I_1}{2^n - 1}$$

Ultima formulă pune în evidență faptul că lucrând cu două valori succesive I_1 și I_2 ce aproximează pe I , se poate obține o a treia valoare presupusă mai bună. Să luăm în continuare în considerare secvența : $I_{0,1}, I_{1,1}, \dots, I_{k,1}, \dots$ obținută prin metoda lui Romberg. Folosind formula anterioară cu $n = 2$, obținem pentru fiecare pereche din șirul de mai sus, o nouă valoare. Să le notăm $I_{0,2}, I_{1,2}, \dots, I_{k,2}, \dots$. Deoa-

rece eroarea este acum proporțională cu h^4 , aplicăm din nou formula ultimei secvențe cu $n = 4$. Se obțin noi valori notate $I_{0,3}, I_{1,3}, \dots, I_{k,3}, \dots$ după care procedura se repetă.

Aplicația 2

Să se aplice metoda lui Romberg și procedura de extrapolare a lui Richardson pentru calculul cu formula trapezelor a integralei $\int_0^{\pi/2} \sin x \, dx$.

Avem:

Nr. puncte	$I_{k,1}$	$I_{k,2}$	$I_{k,3}$	$I_{k,4}$
4	0,987116	1,000008	1,000000	1,000000
8	0,996785	1,000000	1,000000	
16	0,999196	1,000000		
32	0,999799			

Se observă convergența către valoarea exactă 1.

13.3. Metoda dezvoltării în serie a funcției integrant

Așa cum arată numele ei, metoda începe prin dezvoltarea în serie Taylor a funcției integrant, urmată de integrarea seriei astfel obținută. Se exemplifică metoda în exemplul următor:

Aplicația 3

Se cere să se calculeze aproximativ integrala $I = \int_0^b e^{-x^2} \, dx$. Se dezvoltă în serie funcția integrant e^{-x^2} folosind dezvoltarea cunoscută în serie a funcției e^u .
Avem:

$$e^{-x^2} = 1 - x^2 + \frac{x^4}{2} - \frac{x^6}{6} + \frac{x^8}{24} - \frac{x^{10}}{120} + \dots,$$

pentru $x \in \mathbb{R}$.

Funcția integrant fiind continuă, avem în continuare:

$$I = \int_0^b e^{-x^2} dx = \int_0^b 1 \cdot dx - \int_0^b x^2 dx + \int_0^b \frac{x^4}{2} dx - \dots$$

Se obține:

$$I = b - \frac{b^3}{3} + \frac{b^5}{10} - \frac{b^7}{42} + \frac{b^9}{216} - \frac{b^{11}}{1320}$$

Dacă se trunchiază seria după termenul al cincilea, de exemplu, pentru a obține o valoare aproximativă a integralei, eroarea care rezultă (seria fiind alternantă) este în valoarea absolută mai mică decât primul termen neglijat, deci mai mică decât $b^{11}/1320$.

Se observă că pentru valori mari ale lui b , aceasta implică utilizarea unui număr mare de termeni ai seriei.

Este important ca seria obținută să fie rapid convergentă. În caz contrar, se poate încerca mărirea rapidității convergenței seriei, folosind transformarea lui Euler, care înlocuiește seria alternantă

$$u_0 - u_1 + u_2 - u_3 + \dots$$

prin seria cu diferențe finite

$$\sum_{k=0}^{\infty} (-1)^k \frac{\Delta^k u_0}{2^{k+1}} = \frac{1}{2} u_0 - \frac{1}{4} \Delta u_0 + \frac{1}{8} \Delta^2 u_0 - \frac{1}{16} \Delta^3 u_0 + \dots$$

Adeseori a doua serie este mai rapid convergentă.

13.4. Metoda coeficienților nedeterminați

Această metodă este folosită pentru a produce formule de integrare numerică în cazuri speciale, de exemplu în situația în care funcția integrant este oscilantă pe intervalul de integrare. Se urmărește determinarea celor $n + 1$ coeficienți $c_0, c_1, \dots, c_n$, astfel ca:

$$\int_a^b f(x) dx \simeq c_0 f(x_0) + c_1 f(x_1) + \dots + c_n f(x_n)$$

Pentru ca expresia din partea dreaptă a formulei de mai sus să redea cât mai exact valoarea integralei, se impune ca formula să fie verificată și pentru polinoamele : $1, x, x^2, \dots, x^n$, adică :

$$\int_a^b 1 \, dx = c_0 \cdot 1 + c_1 \cdot 1 + \dots + c_n \cdot 1$$

$$\int_a^b x \, dx = c_0 x_0 + c_1 x_1 + \dots + c_n x_n$$

$$\int_a^b x^n \, dx = c_0 x_0^n + c_1 x_1^n + \dots + c_n x_n^n$$

Într-adevăr, în acest caz, formula de integrare numerică va fi verificată pentru orice polinom de grad mai mic sau egal cu n .

$$\begin{aligned} \int_a^b P_n(x) \, dx &= \int_a^b (a_0 + a_1 x + \dots + a_n x^n) \, dx = \\ &= a_0 \int_a^b 1 \, dx + a_1 \int_a^b x \, dx + \dots + a_n \int_a^b x^n \, dx = a_0 \sum_{k=0}^n c_k \cdot 1 + \\ &+ a_1 \sum_{k=0}^n c_k x_k + \dots + a_n \sum_{k=0}^n c_k x_k^n = \sum_{k=0}^n c_k (a_0 + a_1 x_k + \\ &+ \dots + a_n x_k^n) = c_0 P_n(x_0) + c_1 P_n(x_1) + \dots + c_n P_n(x_n) \end{aligned}$$

Aplicația 4

Să se determine coeficienții c_k , astfel ca :

$$\int_{-h}^h y(x) \, dx \simeq h(c_0 y_0 + c_1 y_1 + c_2 y_2)$$

Impunînd conform celor de mai sus ca funcțiile $y(x) = 1, x$ și x^2 să verifice formula, se obține sistemul :

$$2 = c_0 + c_1 + c_2$$

$$0 = -c_0 + c_1$$

$$2/3 = c_0 + c_2$$

cu soluția $c_0 = c_2 = 1/3$, $c_1 = 4/3$. Rezultă :

$$\int_{-h}^h y(x) dx \simeq \frac{h}{3} (y_0 + 4y_1 + y_2)$$

Se recunoaște în expresia de mai sus formula lui Simpson.

Aplicația 5

Să se folosească metoda coeficienților nedeterminați pentru calculul numeric al integralei $\int_0^{2\pi} y(x) \cos x dx$

Să calculăm coeficienții : c_0 , c_1 , c_2 , astfel ca :

$$\int_0^{2\pi} y(x) \cos x dx \simeq c_0 y(0) + c_1 y(\pi) + c_2 y(2\pi)$$

Impunem ca formula de mai sus să fie verificată pentru : $y(x) = 1$, x și x^2

$$0 = c_0 + c_1 + c_2$$

$$0 = \pi c_1 + 2\pi c_2$$

$$4\pi = \pi^2 c_1 + 4\pi^2 c_2$$

Rezultă soluția : $c_0 = 1/2$, $c_1 = -1/\pi$, $c_2 = 1/2\pi$. Deci :

$$\int_0^{2\pi} y(x) \cos x dx \simeq \frac{1}{2} y(0) - \frac{1}{\pi} y(\pi) + \frac{1}{2\pi} y(2\pi)$$

13.5. Integrarea numerică a integralelor improprii

Pentru calculul numeric al integralelor improprii convergente se pot da o diversitate de proceduri puse în evidență în aplicațiile următoare :

Aplicația 6

Să se calculeze integrala improprie $I = \int_0^1 \frac{e^x dx}{\sqrt{x}}$.

Se observă că funcția integrant ia valoare infinită în limita inferioară de integrare.

Efectuând o schimbare de variabilă $u = \sqrt{x}$, se obține :

$$I = 2 \int_0^1 e^{u^2} du.$$

Dezvoltând în serie de puteri funcția integrant și integrând termen cu termen seria rezultată, se obține I ca sumă a unei serii numerice convergente :

$$I = 2 \int_0^1 \left(1 + \frac{u^2}{1!} + \frac{u^4}{2!} + \dots \right) du = 2 \left(1 + \frac{1}{3} + \frac{1}{10} + \dots \right)$$

Aplicația 7

Să se calculeze integrala improprie $I = \int_0^1 \frac{dx}{(1+x)\sqrt{x}}$

Se observă că integrala are limită superioară infinită ; în plus, funcția integrant devine nemărginită în limita inferioară de integrare.

Folosind proprietatea de aditivitate a integralei definite în raport cu domeniul, se obține :

$$I = \int_0^{\infty} \frac{dx}{(1+x)\sqrt{x}} = \int_0^1 \frac{dx}{(1+x)\sqrt{x}} + \int_1^{\infty} \frac{dx}{(1+x)\sqrt{x}}$$

Efectuând în a doua integrală schimbarea de variabilă $x = 1/u$, $dx = -du/u^2$, rezultă :

$$I = \int_0^1 \frac{dx}{(1+x)\sqrt{x}} + \int_0^1 \frac{du}{(1+u)\sqrt{u}} = 2 \int_0^1 \frac{dx}{(1+x)\sqrt{x}}$$

Deci limita superioară infinită a fost evitată. Pentru a evita în continuare ca funcția să devină nemărginită în limita inferioară vom integra prin părți.

Se obține :

$$I = 2 \int_0^1 \frac{dx}{(1+x)\sqrt{x}} = 4 \sqrt{x} \frac{1}{1+x} \Big|_0^1 + 4 \int_0^1 \frac{\sqrt{x} dx}{(1+x)^2} =$$

$$= 2 + 4 \int_0^1 \frac{\sqrt{x} dx}{(1+x)^2}$$

Ultima integrală poate fi integrată folosind de exemplu o formulă de cvadratură.

Aplicația 8

Să se calculeze integrala proprie $I = \int_0^{\infty} e^{-x^2-1/x^2} dx$.

Folosind un parametru n se formează funcția :

$$I(u) = \int_0^{\infty} e^{-x^2-u^2/x^2} dx.$$

Presupunind că se poate deriva sub semnul integral, avem :

$$I'(n) = -2u \int_0^{\infty} \frac{1}{x^2} e^{-x^2-u^2/x^2} dx.$$

Folosind schimbarea de variabilă $y = u/x$, se obține :

$$I'(u) = -2 \int_0^{\infty} e^{-y^2-u^2/y^2} dy = -2I(u)$$

Ecuația diferențială de ordinul întâi rezultată are soluția generală $I(u) = Ce^{-2u}$.

Se determină constanta C folosind faptul că integrala

$$I(0) = \int_0^{\infty} e^{-x^2} dx$$

care intervine în calculul probabilităților are valoarea $\sqrt{\pi}/2$.

Deci

$$I(0) = \sqrt{\pi}/2 = C$$

Rezultă :

$$\int_0^{\infty} e^{-x^2 - u^2/x^2} dx = \frac{\sqrt{\pi}}{2} e^{-2u}$$

În final, pentru valoarea $u = 1$, se obține valoarea cerută a integralei

$$I = \int_0^{\infty} e^{-x^2 - 1/x^2} dx = \sqrt{\pi}/2e^2.$$

13.6. Integrarea numerică a integralei nedefinite și a integralelor multiple

Fie funcția f integrabilă pe $[a, b]$. Oricare ar fi $x \in [a, b]$, funcția f este integrabilă pe $[a, x]$. Relația

$$F(x) = \int_a^x f(t) dt, \quad x \in [a, b]$$

definește o funcție numită *integrală nedefinită a funcției f* . Să presupunem că funcția integrant este dată printr-o tabelă de valori de forma :

$$\begin{array}{c|cccc} x_k & a = x_0 & x_1 & x_2 \dots x_n & b \\ \hline f(x_k) & y_0 & y_1 & y_2 & y_n \end{array}$$

avînd pasul constant h .

Folosind formula trapezelor de exemplu, este evident că

$$\int_{x_k}^{x_{k+1}} f(t) dt = F(x_{k+1}) - F(x_k) \simeq \frac{h}{2} (y_k + y_{k+1})$$

Deci

$$F(x_{k+1}) \simeq F(x_k) + \frac{h}{2} (y_k + y_{k+1})$$

Se obține deci o formulă care permite calculul iterativ aproximativ de forma unei tabele de valori al funcției $F(x)$.

Se consideră în continuare problema determinării volumului mărginit de o suprafață dată $z = f(x, y)$ situată deasupra planului xOy (mai precis, ale cărei puncte au cota z pozitivă) și de patru plane ale căror ecuații sunt respectiv $x = a$, $x = b$, $y = c$, $y = d$, unde $a < b$, $c < d$. În acest caz volumul se exprimă prin integrala dublă :

$$V = \iint_{\Delta} f(x, y) dx dy = \int_a^b dx \int_c^d f(x, y) dy$$

Să calculăm numeric integrala dublă de mai sus. Din punct de vedere geometric $\int_c^d f(x, y) dy = S(x)$ reprezintă mărimea ariei secțiunii volumului dat cu un plan perpendicular pe axa Ox dus la distanța x de origine, $x \in [a, b]$.

Evident

$$V = \int_a^b S(x) dx,$$

adică se calculează printr-o integrală definită simplă, ce poate fi integrată numeric, de exemplu folosind formula trapezelor.

Efectuînd o diviziune a intervalului $[a, b]$ cu noduri echidistante

$$a = x_0 < x_1 < x_2 < \dots < x_n = b$$

se obține, folosind formula trapezelor :

$$V \simeq \frac{b-a}{2n} [S(x_0) + 2S(x_1) + \dots + 2S(x_{n-1}) + S(x_n)]$$

Pentru calculul valorilor $S(x_k)$ ne folosim de relația de definiție a lui $S(x)$. Avem :

$$S(x_k) = \int_c^d f(x_k, y) dy.$$

Deoarece mărimea x_k în expresia funcția integrant reprezintă o constantă, urmează că și această integrală poate fi integrată cu ajutorul unei formule de cvadratură, de exemplu

folosind formula trapezelor. Folosind o diviziune a intervalului $[c, d]$ cu noduri echidistante, se obține :

$$c = y_0 < y_1 < y_2 < \dots < y_n = d$$

$$S(x_k) \simeq \frac{d-c}{2n} [(f(x_k, y_0) + 2f(x_k, y_1) + \dots + 2f(x_k, y_{n-1}) + f(x_k, y_n)] \quad k = 0, 1, 2, \dots, n$$

Aplicația 9

Se prezintă subrutina de calcul intitulată *TRAPEZ* pentru calculul aproximativ al integralei $\int_{x_1}^{x_n} f(x) dx$ folosind formula trapezelor.

Parametrii fictivi ai subrutinei sînt :

N — numărul de puncte în care se calculează funcția $f(x)$
 S — valoarea aproximativă calculată a integralei

5 | 6 | 7

```

SUBROUTINE TRAPEZ (N, S)
DIMENSION X(N)
H = (X(N) - X(1)) / (N - 1)
S = 0.
NI = N - 1
DO 1 K = 2, NI
X(K) = X(1) + (K - 1) * H
S = S + H * F(X(K))
1 CONTINUE
S = S + (H/2) * (F(X(1)) + F(X(BN)))
WRITE (108, 2) S
2 FORMAT (50X, 'INTEGRALA=', G 18.8)
RETURN
END

```

În cadrul subrutinei, F reprezintă funcția integrant $f(x)$ și trebuie scrisă conform regulilor referitoare la funcțiile aritmetice.

14. INTEGRAREA NUMERICĂ A ECUAȚIEI DIFERENȚIALE DE ORDINUL ÎNTII CU CONDIȚII ÎNȚIALE

14.1. Introducere

Fie ecuația diferențială de ordinul întâi $y' = f(x, y)$ cu condiția inițială : $y(x_0) = y_0$.

Se presupune în cele ce urmează că soluția exactă a ecuației enunțate există și este unică. Pentru aceasta este suficient ca funcția $f(x, y)$ să fie continuă, iar derivata sa în raport cu x să existe și să fie mărginită.

O metodă numerică pentru integrarea ecuației diferențiale este o procedură ce produce valori aproximative ale soluției exacte în anumite puncte, folosind numai operații de adunare, scădere, înmulțire și împărțire, de asemenea evaluări de funcții. Deci, se caută soluția ecuației sub forma unei tabele de valori cu pas constant :

x_k		x_0	x_1	$x_2 \dots x_n$
y_k		y_0	y_1	$y_2 \dots y_n$

Din această tabelă de valori se cunoaște pentru început numai cuplul de valori x_0, y_0 dat prin condiția inițială și care determină coordonatele punctului inițial $M_0 (x_0, y_0)$, punct de plecare al curbei integrale căutate. Folosind punctul M_0 și una din metodele prezentate în continuare, se pot calcula cu aproximație și celelalte puncte $M_1 (x_1, y_1), M_2 (x_2, y_2), \dots, M_n (x_n, y_n)$ prin care trece curba integrală.

14.2. Metoda dezvoltării în serie Taylor

Să presupunem că soluția $y(x)$ a ecuației diferențiale se poate dezvolta în serie Taylor în vecinătatea fiecărui punct x_k .

Avem :

$$y(x) = y(x_k) + \frac{x - x_k}{1!} y'(x_k) + \frac{(x - x_k)^2}{2!} y''(x_k) + \dots$$

Dacă se presupune cunoscută valoarea lui $y(x)$ în punctul x_k , folosind funcția $f(x, y)$ definită prin ecuația diferențială, putem calcula în funcție de aceste valori definite în punctul x_k derivata de orice ordin pentru $y(x)$. Într-adevăr, avem :

$$y'(x_k) = f(x, y) \Big|_{\substack{x=x_k \\ y=y(x_k)}}$$

$$y''(x_k) = f'_x(x, y) + y'(x) f'_y(x, y) \Big|_{\substack{x=x_k \\ y=y(x_k)}} = f'_x(x, y) + \\ + f(x, y) f''_y(x, y) \Big|_{\substack{x=x_k \\ y=y(x_k)}}$$

Introducând valorile obținute pentru derivate în dezvoltarea în serie și notînd : $x = x_{k+1}$, $x_{k+1} - x_k = h$, avem :

$$y(x_{k+1}) = y(x_k) + \frac{h}{1!} f[x_k, y(x_k)] + \frac{h^2}{2} \{f'_x(x_k, y(x_k)) + \\ + f(x_k, y(x_k)) f'_y(x_k, y(x_k))\} + \dots$$

Pentru $k = 0$, obținem :

$$y(x_1) = y_0 + \frac{h}{1!} f(x_0, y_0) + \frac{h^2}{2!} [f'_x(x_0, y_0) + \\ + f(x_0, y_0) f'_y(x_0, y_0)] + \dots$$

Dacă se vor lua în formula anterioară un număr suficient de termeni, se poate calcula aproximativ $y(x_1)$ notată cu y_1 în tabela de valori. Folosind x_1 și valoarea y_1 astfel calculată, se poate obține valoarea y_2 și în acest mod se poate produce în mod iterativ tot șirul de valori cerute

$y_0, y_1, \dots, y_n$. Cu alte cuvinte se poate pune în evidență un algoritm de calcul de forma :

$$y_{k+1} = y_k + hf(x_k, y_k) + \frac{h^2}{2} [f'_x(x_k, y_k) + \\ + f(x_k, y_k) f'_y(x_k, y_k)] + \dots \\ k = 0, 1, 2, \dots, n - 1$$

Metoda dezvoltării în serie Taylor este considerată o metodă directă în sensul în care pentru determinarea punctului $M_{k+1}(x_{k+1}, y_{k+1})$ sînt necesare numai informații despre punctul anterior $M_k(x_k, y_k)$. Deoarece adeseori este complicat sau chiar imposibil de a evalua pe $y', y'', y''', \dots$, metoda dezvoltării în serie Taylor are o aplicativitate redusă, servind mai mult drept criteriu de apreciere pentru celelalte metode. În acest sens, dacă folosind o metodă oarecare se calculează o parte a seriei Taylor ce conține termeni în care puterea maximă a lui h este n , se spune că metoda respectivă este de ordinul n . Dacă în algoritmul de calcul pentru dezvoltarea în serie Taylor ne limităm numai la primii doi termeni, se obține :

$$y_{k+1} = y_k + hf(x_k, y_k), \quad k = 0, 1, 2, \dots, n - 1$$

Acest algoritm redus poartă numele de *formula lui Euler*. Evident că limitînd atît de drastic dezvoltarea în serie Taylor, formula lui Euler nu poate furniza de obicei o precizie satisfăcătoare, în schimb permite o interpretare geometrică pentru construirea soluției (fig. 14.1).

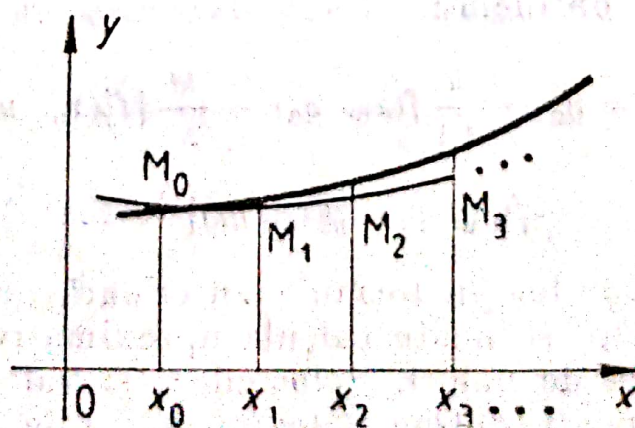


Fig. 14.1

Într-adevăr, din punct de vedere geometric, formula lui Euler constă în a înlocui curba integrală $y(x)$, care trece prin punctul inițial M_0 , printr-un contur poligonal $M_0M_1M_2M_3 \dots M_n$ construit astfel: începem prin a construi o dreaptă ce trece prin punctul $M_0(x_0, y_0)$ și de coeficient unghiular cunoscut: $m_0 = y'_0 = f(x_0, y_0)$. Se obține astfel punctul $M_1(x_1, y_1)$ la intersecția drepte cu ordonata ridicată în punctul x_1 . În continuare se construiește o dreaptă ce trece prin punctul M_1 astfel calculat, de coeficient unghiular $m_1 = y'_1 = f(x_1, y_1)$. Se obține punctul M_2 la intersecția acestei drepte cu ordonata în punctul x_2 . Procedura se repetă pentru determinarea în același mod a punctelor $M_3, M_4, \dots, M_n$. Cu cât crește distanța între x_k și x_0 , cu atât descrește precizia rezultatelor furnizate de formula lui Euler, ca de altfel și a algoritmului dezvoltării în serie Taylor.

Din punct de vedere al preciziei, formula lui Euler impune o metodă de ordinul 1, deoarece folosește o parte ce conține doi termeni din dezvoltarea în serie Taylor.

14.3. Metoda Runge-Kutta

Metodele Runge-Kutta își propun să evite calculul derivatelor funcției $f(x, y)$, care intervin în algoritmul dezvoltării în serie Taylor, înlocuind aceste derivate prin evaluări ale funcției $f(x, y)$ în puncte intermediare. Se urmărește formarea de algoritme de tipul:

$$y_{i+1} = y_i + \sum_{j=1}^n \alpha_j \cdot K_j, \quad \alpha_j \in \mathbf{R},$$

unde coeficienții K_j se determină cu ajutorul valorilor funcției f calculată în anumite puncte intermediare din intervalul (x_i, x_{i+1}) , astfel încât dezvoltarea din partea dreaptă în formula de mai sus să coincidă cu dezvoltarea în serie Taylor a lui y_{i+1} pînă la termenii de rang n . Se obțin astfel diverse formule de calcul, dintre acestea cel mai des utilizate în practică fiind următoarele: pentru $n = 3$, se obține algoritmul metodei Runge-Kutta de ordinul 3.

$$y_{i+1} = y_i + \frac{1}{3} (K_1 + 4K_2 + K_3)$$

$$K_1 = hf(x_i, y_i)$$

$$K_2 = hf\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_1\right)$$

$$K_3 = hf(x_i + h, y_i - K_1 + 2K_2)$$

Pentru $n = 4$, se obține algoritmul metodei Runge-Kutta de ordinul 4, frecvent utilizat în aplicații.

$$y_{i+1} = y_i + \frac{1}{6} (K_1 + 2K_2 + 2K_3 + K_4)$$

$$K_1 = hf(x_i, y_i)$$

$$K_2 = hf\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_1\right)$$

$$K_3 = hf\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_2\right)$$

$$K_4 = hf(x_i + h, y_i + K_3)$$

Metodele Runge-Kutta sînt metode directe în sensul că pentru a calcula pe y_{k+1} avem nevoie numai de informații ce intervin la pasul precedent unde este calculat punctul (x_k, y_k) . Aceste metode prezintă o bună stabilitate, se poate schimba pasul de iterare în cursul iterației, construcția unui program de calcul pentru calculator pentru aceste metode nu ridică probleme. Singurul dezavantaj al acestor metode constă în necesitatea de a evalua funcția ce intervine în membrul drept al ecuației diferențiale, de n ori pentru o metodă de ordinul n la fiecare pas, ceea ce mărește destul de mult timpul de calcul.

Aplicația 1

Fiind dată ecuația $y' = y$ cu condiția $y(0) = 1$, se cere să se determine valoarea lui y în punctul $x = 0,1$ folosind metoda Runge-Kutta de ordinul patru.

Folosind algoritmul prezentat anterior cu un pas h egal cu 0,1, se obține :

$$i = 0, h = 0,1, x_0 = 0, y_0 = 1$$

$$K_1 = hf(x_0, y_0) = hy_0 = 0,1 \cdot 1 = 0,1$$

$$K_2 = hf(x_0 + h/2, y_0 + K_1/2) = h(y_0 + K_1/2) = 0,105$$

$$K_3 = hf(x_0 + h/2, y_0 + K_2/2) = h(y_0 + K_2/2) = 0,10525$$

$$K_4 = hf(x_0 + h, y_0 + K_3) = h(y_0 + K_3) = 0,110525$$

$$\text{deci } y(0,1) = y_1 = y_0 + (K_1 + 2K_2 + 2K_3 + K_4)/6 = \\ = 1,1051708$$

Ecuatia dată admite soluția exactă $y = e^x$. Rezultă $y(0,1) = 1,1051709$, deci o valoare care diferă cu o unitate la cea de-a șaptea zecimală față de valoarea calculată aproximativ prin metoda Runge-Kutta.

14.4. Metode predictor — corector

Aceste metode, așa cum arată și numele lor, folosesc două formule și anume :

- o *formulă de prezicere* cu care se obține o primă estimare pentru valoarea lui y_k ;
- o *formulă de corectie*, mai precisă, cu ajutorul căreia valoarea lui y_k obținută în urma etapei de prezicere, este îmbunătățită.

Metoda Euler modificată folosește următoarea pereche de formule de precizie și corectare :

$$y_{k+1} = y_k + hy'_k,$$

$$y_{k+1} = y_k + \frac{1}{2} h(y'_k + y'_{k+1})$$

Formula de prezicere se recunoaște a fi formula lui Euler obținută prin trunchierea seriei Taylor la primii doi termeni.

A doua formulă, și anume formula de prezicere, este obținută observînd că :

$$y_{k+1} - y_k = \int_{x_k}^{x_{k+1}} y' \, dx$$

și aproximînd integrala prin formula trapezelor :

$$\int_{x_k}^{x_{k+1}} y' \, dx \simeq \frac{h}{2}(y'_k + y'_{k+1})$$

Deoarece $y'_k = f(x_k, y_k)$, formula de prezicere permite o primă evaluare a lui y_{k+1} . Apoi această valoare de prezicere este utilizată în membrul drept al formulei de corecție, ținînd cont că $y'_{k+1} = f(x_{k+1}, y_{k+1})$ pentru a obține în partea stîngă a semnului egal o valoare y_{k+1} corectată. În continuare, această valoare corectată este folosită din nou în partea dreaptă a formulei de corecție pentru a obține în partea stîngă a semnului egal o nouă corecție. Procesul se repetă pînă cînd diferența dintre două corecții succesive devine mai mică decît o valoare dată. Metoda Euler modificată este o metodă de ordinul doi. Din punct de vedere geometric, formula corectoare din metoda Euler modificată permite aproximarea unei curbe integrale printr-un contur poligonal, luînd ca pantă pentru fiecare segment media aritmetică a pantelor calculate cu formula lui Euler pentru două puncte succesive.

Metoda Milne folosește următoarea pereche de formule de prezicere și corectare :

$$y_{k+1} = y_{k-3} + (4h/3)(2y'_{k-2} - y'_{k-1} + 2y'_k)$$

$$y_{k+1} = y_{k-1} + (h/3)(y'_{k+1} + 4y'_k + y'_{k-1})$$

Formula corectoare a fost obținută în același fel ca și în cazul metodei Euler modificate, dar utilizînd o integrare numerică după formula lui Simpson. Metoda lui Milne impune pentru aplicarea formulei de prezicere, din start cunoașterea a patru valori $y_k, y_{k-1}, y_{k-2}, y_{k-3}$ care trebuie obținute aplicînd o altă metodă, de obicei o metodă directă. Se poate arăta că metoda Milne este o metodă de ordinul patru.

Metoda Adams folosește următoarea pereche de formule de prezicere și corectare :

$$y_{k+1} = y_k + (h/24)(55y'_k - 59y'_{k-1} + 37y'_{k-2} - 9y'_{k-3})$$

$$y_{k+1} = y_k + (h/24)(9y'_{k+1} + 19y'_k - 5y'_{k-1} + y'_{k+2})$$

Și în cazul metodei lui Adams sînt necesare cunoașterea din start a patru valori : y_0, y_1, y_2, y_3 . Ca și metoda Milne, metoda lui Adams este de ordinul patru. Metodele tip predictor-corrector, folosind același număr de noduri ca și metodele directe dar utilizînd mai rațional valorile deja calculate pentru soluție, oferă o precizie mai bună.

14.5. Metoda Picard

Această metodă permite calcularea unei soluții aproximative de formă analitică.

Fie ecuația : $y' = f(x, y)$, cu condiția inițială $y(x_0) = y_0$. Presupunem că funcția $f(x, y)$ este continuă într-un domeniu din planul xOy care conține punctul $M(x_0, y_0)$. Presupunem în plus că derivata $f'_y(x, y)$ a funcției $f(x, y)$ este mărginită în acest domeniu. Integrînd ambii membri ai ecuației diferențiale, obținem :

$$\int_{x_0}^x y' dx = \int_{x_0}^x f([u, y(u)] du$$

sau

$$y(x) - y_0 = \int_{x_0}^x f(u, y) du$$

Se înlocuiește deci ecuația diferențială cu o ecuație integrală.

Dacă se ia ca o primă aproximație a soluției funcția $y_1(x) = y_0$, atunci o soluție mai bună va fi funcția $y_2(x)$ obținută din formula de mai sus, înlocuind în partea dreaptă a semnelui egal pe y cu y_1 .

Avem :

$$y_2(x) = y_0 + \int_{x_0}^x f([u, y_1(u)] du$$

Prin repetarea acestui proces obținem formula iterativă

$$y_k(x) = y_0 + \int_{x_0}^x f(u, y_{k-1}(u)) du$$

Șirul de funcții astfel produs :

$$y_1(x), y_2(x), \dots, y_k(x), \dots$$

este convergent către funcția $y(x)$, soluție a ecuației diferențiale. Deficiența metodei aproximațiilor succesive constă în faptul că de obicei apar greutăți la calculul integralelor ce intervin în algoritmul de calcul.

Aplicația 2

Se dă ecuația $y' = 2x + y$ și condiția inițială $y(0) = 1$. Să se determine o soluție analitică aproximativă, folosind metoda lui Picard. Integrând ambii termeni ai ecuației, avem :

$$y(x) = 1 + \int_0^x (2u + y) du$$

Luind pe $y_1 = 1$ ca primă aproximație, găsim o a doua aproximație pentru soluția ecuației de forma :

$$y_2(x) = 1 + \int_0^x (2u + 1) du = 1 + x + x^2$$

A treia aproximație va fi :

$$y_3(x) = 1 + \int_0^x (2u + 1 + u + u^2) du = 1 + x + \frac{3}{2}x^2 + \frac{x^3}{3}$$

Aproximații de ordin superior sînt obținute în același mod.

14.6. Studiul erorilor, metode relativ stabile

Metodele numerice introduc erori care pot fi evaluate folosind seria Taylor. Astfel, formula lui Euler reprezentînd o trunchiere a seriei Taylor după primii doi termeni, intro-

duce o eroare locală de trunchiere estimată după primul termen care lipsește din seria Taylor și anume termenul al treilea :

$$\epsilon_T \approx \frac{h^2}{2} y_k''$$

Pentru alte metode calculul erorii locale de trunchiere este mai dificil.

De exemplu, în cazul formulei de prezicere din metoda Milne, $y_{k+1} = y_{k-3} + (4h/3)(2y_{k-2}' - y_{k-1}' + 2y_k')$ calculăm eroarea astfel : pentru $k = 1$, formula de prezicere devine :

$$y_2 - y_{-2} = (4h/3)(2y_{-1}' - y_0' + 2y_1').$$

Folosind dezvoltarea în serie Taylor în jurul punctului x_0 observăm că

$$x_k - x_0 = kh$$

$$y_k = y_0 + (kh)y_0' + \frac{1}{2}(kh)^2 y_0^{(2)} + \frac{1}{6}(kh)^3 y_0^{(3)} + \\ + \frac{1}{24}(kh)^4 y_0^{(4)} + \frac{1}{120}(kh)^5 y_0^{(5)} + \dots$$

Luind pentru indicele k valorile 2 și -2 în formula de mai sus, obținem :

$$y_2 - y_{-2} = 4hy_0' + \frac{8}{3}h^3 y_0^{(3)} + \frac{8}{15}h^5 y_0^{(5)} + \dots$$

Derivând formula lui Taylor, avem :

$$y_k' = y_0' + (kh)y_0^{(2)} + \frac{1}{2}(kh)^2 y_0^{(3)} + \frac{1}{6}(kh)^3 y_0^{(4)} + \\ + \frac{1}{24}(kh)^4 y_0^{(5)} + \dots$$

Folosind această expresie se poate estima expresia : $2y_{-1}' - y_0' + 2y_1'$ care intervine în partea dreaptă a formulei de prezicere de tip Milne.

$$2y_{-1}' - y_0' + 2y_1' = 3y_0' + 2h^2 y_0^{(3)} + \frac{1}{6}h^4 y_0^{(5)} + \dots$$

Eroarea locală de trunchiere este prin urmare :

$$(y_2 - y_{-2}) - \frac{4}{3} h(2y'_{-1} - y'_0 + 2y'_1) = \frac{14}{45} h^5 y^{(5)}_0 + \dots$$

Folosind pentru estimare primul termen al expresiei din partea dreaptă a semnelui egal, se obține : $\varepsilon_T \simeq \frac{14}{45} h^5 y^{(5)}_{k-1}$

fapt care confirmă că metoda Milne este de ordinul patru.

Este evident că se poate mări precizia metodelor descrise, micșorând pasul h dar și aceasta până la o anumită limită, deoarece micșorarea valorii pasului are ca efect mărirea numărului de pași și deci mărirea erorilor de rotunjire. Pe de altă parte, erorile de trunchiere locale se cumulează în metodele de integrare, pas cu pas, pe de o parte fiindcă punctul ce constituie date de intrare pentru calculele pasului respectiv este incorect, pe de altă parte, fiecare pas introduce o eroare suplimentară. Astfel, dacă se pleacă din punctul i cu o valoare aproximativă Y_i a lui y_i și dacă se ține seama de erorile de cumulare în punctul $i + k$, eroarea cumulată va fi $\varepsilon_{i+k} = y_{i+k} - Y_{i+k}$. Se spune că metoda de integrare este stabilă dacă eroarea în $i + k$ rămâne mărginită atunci când $k \rightarrow \infty$.

Mai semnificativă decât eroarea absolută este eroarea relativă definită ca raportul dintre eroare și soluția exactă. Se spune că metoda de integrare este relativ stabilă dacă eroarea relativă în $i + k$ este mărginită când $k \rightarrow \infty$. Pentru estimarea erorii relative introduse de o metodă numerică se folosește de obicei ecuația test : $y' = Ay$, ($A = \text{constant}$) cu condiția inițială : $y(0) = 1$. Soluția exactă a acestei ecuații, cu condiție inițială, este $y(x) = e^{Ax}$. Să examinăm comportarea algoritmului Euler.

Avem :

$$Y_{k+1} = Y_k + hf(x_k, Y_k) = (1 + Ah) \cdot Y_k$$

Soluția exactă, pe de altă parte, se obține folosind dezvoltarea în serie Taylor de forma :

$$y_{k+1} = y_k + \frac{h}{1!} y'_k + \varepsilon_T = y_k(1 + Ah) + \varepsilon_T,$$

unde ε_r reprezintă eroarea de trunchiere. Neglijînd eroarea de trunchiere și scăzînd cele două relații termen cu termen, avem :

$$\varepsilon_{k+1} \simeq y_{k+1} - Y_{k+1} = (1 + Ah)\varepsilon_k$$

Iterînd relația de mai sus, se obține :

$$\varepsilon_1 \simeq (1 + Ah)\varepsilon_0$$

$$\varepsilon_2 \simeq (1 + Ah)\varepsilon_1 \simeq (1 + Ah)^2\varepsilon_0$$

$$\varepsilon_k \simeq (1 + Ah)^k\varepsilon_0$$

Metoda lui Euler va fi deci stabilă dacă $|1 + Ah| < 1$, adică $-2 < Ah < 0$.

Deoarece soluția exactă a ecuației test este cunoscută și anume $y(x) = e^{Ax}$ sau pentru $x = kh$, $y_k = y(kh) = e^{Akh}$, putem forma eroarea relativă :

$$\varepsilon_k^r = \frac{\varepsilon_k}{y_k} = \varepsilon_0 \frac{(1 + Ah)^k}{e^{Akh}} = \varepsilon_0 \frac{1 + kAh + \frac{1}{2}k(k-1)A^2h^2 + \dots}{1 + kAh + \frac{1}{2}k^2A^2h^2 + \dots} \leq \varepsilon_0$$

Rezultă $\varepsilon_k^r \leq \varepsilon_0$, adică stabilitatea relativă a metodei Euler este asigurată pentru orice A . Procedînd în același mod, se găsește că pentru ecuația test dată, metodele Euler îmbunătățită, Taylor și Adams se dovedesc a fi relativ stabile. În schimb metoda Milne se dovedește instabilă cînd constanta A este negativă. Această metodă nu este recomandabilă pentru ecuațiile avînd ca soluții funcții descrescătoare.

Aplicația 3

Se prezintă subrutina de calcul intitulată *RUNGE* pentru calculul aproximativ al soluției unei ecuații diferențiale de ordinul întâi cu condiție inițială după metoda Runge-Kutta de ordinul 4. Parametrii fictivi ai subrutinei sînt :

X1 — valoarea inițială dată pentru argumentul x

Y1 — valoarea inițială dată pentru funcția soluție calculată în punctul *X1*

- N** — numărul de puncte în care se calculează funcția soluție
X — vector conținând valorile argumentului x
Y — vector conținând valorile funcției soluție
H — valoarea pasului ales pentru efectuarea calculelor

5 | 6 | 7

```

SUBROUTINE RUNGE (X1, Y1, N, H, X, Y)
DIMENSION X(1000), Y(1000)
X(1) = X1
Y(1) = Y1
NC = N-1
DO 1 K = 1, NC
  K1 = H * F(X(K), Y(K))
  K2 = H * F(X(K) + H/2, Y(K) + K1/2)
  K3 = H * F(X(K) + H/2, Y(K) + K2/2)
  K4 = H * F(X(K) + H, Y(K) + K3)
  Y(K+1) = Y(K) + (K1 + 2 * K2 + 2 * K3 + K4)/6
1  X(K+1) = X(K) + H
2  WRITE (108, 2) (Y(K), K = 1, N)
  FORMAT (5X, 100 G12,6)
RETURN
END
  
```

În cadrul subrutinei, F reprezintă funcția $f(x, y)$ a ecuației $y' = f(x, y)$ și trebuie scrisă conform regulilor referitoare la funcțiile aritmetice.

15. INTEGRAREA NUMERICĂ A SISTEMELOR DE ECUAȚII DIFERENȚIALE DE ORDINUL ÎNTÎI CU CONDIȚII ÎNȚIALE

15.1. Introducere

Aceste sisteme joacă un rol important, deoarece se poate arăta ușor, că orice ecuație diferențială de ordin superior, scrisă sub forma normală, poate fi redusă la un astfel de sistem. Fie de exemplu ecuația :

$y'' = g(x, y, y')$ cu condițiile inițiale : $y(x_0) = y_0, y'(x_0) = z_0$.
Cu notația $y' = z$, această ecuație se reduce la sistemul :
 $y' = z, z' = g(x, y, z)$ cu condițiile inițiale : $y(x_0) = y_0, z(x_0) = z_0$.

Mai mult, procedînd în același mod, se poate arăta că și sistemele de ecuații de ordin superior pot fi reduse la sisteme de ordinul întii. Fie de exemplu sistemul :

$$F(t, x, x', x'', y, y', y'') = 0$$

$$G(t, x, x', x'', y, y', y'') = 0$$

cu condițiile inițiale :

$$x(t_0) = x_0, \quad x'(t_0) = u_0$$

$$y(t_0) = y_0, \quad y'(t_0) = v_0$$

Notînd : $x' = u, y' = v$, obținem sistemul :

$$x' = u$$

$$y' = v$$

$$F(t, x, u, u', y, v, v') = 0$$

$$G(t, x, u, u', y, v, v') = 0$$

cu condițiile inițiale :

$$\begin{aligned}x(t_0) &= x_0, & u(t_0) &= u_0 \\y(t_0) &= y_0, & v(t_0) &= v_0\end{aligned}$$

Metodele de integrare numerică a ecuațiilor diferențiale de ordinul întâi se pot extinde cu ușurință și pentru integrarea sistemelor de ecuații de ordinul întâi.

15.2. Algoritme de calcul

Fie sistemul de ecuații diferențiale de ordinul unu :

$$\begin{aligned}y' &= f(x, y, z) \\z' &= g(x, y, z)\end{aligned}$$

cu condițiile inițiale : $y(x_0) = y_0, z(x_0) = z_0$

Se dau în continuare algoritmele de calcul pentru rezolvarea sistemelor prin metodele Euler și Runge-Kutta de ordinul 4.

Metoda Euler

$$y_{i+1} = y_i + hf(x_i, y_i, z_i)$$

$$z_{i+1} = z_i + hg(x_i, y_i, z_i), \quad i = 0, 1, 2, \dots$$

Metoda Runge — Kutta de ordinul 4

$$y_{i+1} = y_i + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4)$$

$$z_{i+1} = z_i + \frac{1}{6}(L_1 + 2L_2 + 2L_3 + L_4)$$

unde :

$$K_1 = hf(x_i, y_i, z_i)$$

$$L_1 = hg(x_i, y_i, z_i)$$

$$K_2 = hf\left(x_i + \frac{h}{2}, y_i + \frac{K_1}{2}, z_i + \frac{L_1}{2}\right)$$

$$L_2 = hg \left(x_i + \frac{h}{2}, y_i + \frac{K_1}{2}, z_i + \frac{L_1}{2} \right)$$

$$K_3 = hf \left(x_i + \frac{h}{2}, y_i + \frac{K_2}{2}, z_i + \frac{L_2}{2} \right)$$

$$L_3 = hg \left(x_i + \frac{h}{2}, y_i + \frac{K_2}{2}, z_i + \frac{L_2}{2} \right)$$

$$K_4 = hf(x_i + h, y_i + K_3, z_i + L_3)$$

$$L_4 = hg(x_i + h, y_i + K_3, z_i + L_3)$$

Aplicația 1

Se dă sistemul de ecuații diferențiale $y' = z$, $z' = y$ cu condițiile inițiale $y(0) = 0$, $z(0) = 1$.

Folosind metoda Runge-Kutta de ordinul 4 să se calculeze soluția sistemului în punctul $x = 0,1$.

Avem $f(x, y, z) = z$, $g(x, y, z) = y$, $x_0 = 0$, $z_0 = 1$

Se alege pasul $h = 0,1$.

Folosind algoritmul de calcul dat mai sus se obține :

$$K_1 = 0,1 \cdot 1 = 0,1 \quad K_3 = 0,1 \cdot 1,0025 = 0,10025$$

$$L_1 = 0,1 \cdot 0 = 0 \quad L_3 = 0,1 \cdot 0,05 = 0,005$$

$$K_2 = 0,1 \cdot 1 = 0,1 \quad K_4 = 0,1 \cdot 1,005 = 0,1005$$

$$L_2 = 0,1 \cdot 0,05 = 0,005 \quad L_4 = 0,1 \cdot 0,10025 = 0,010025$$

Rezultă :

$$y_1 = 0 + (0,1 + 0,2 + 0,1005 + 0,1005)/6 = 0,0835$$

$$z_1 = 1 + (0 + 0,01 + 0,01 + 0,010025)/6 = 1,005004$$

Dar pentru sistemul de ecuații dat soluția exactă este $y = 1/2e^x - 1/2e^{-x}$, $z = 1/2e^x + 1/2e^{-x}$, deci pentru $x = 0,1$ se obțin valorile $y = 0,1001667$, $z = 1,0050041$.

Rezultă o eroare totală mai mică de două sutimi.

Metoda Adams

Presupunem că avem la dispoziție valorile de start $y_0, y_1, y_2, y_3, z_0, z_1, z_2, z_3$. De fapt y_0 și z_0 sunt cunoscute

din condițiile inițiale. Celelalte valori trebuie obținute folosind o metodă directă, de exemplu Runge-Kutta.

În continuare formulele de prezicere, sînt de forma :

$$y_{k+1} = y_k + (h/24)(55y'_k - 59y'_{k-1} + 37y'_{k-2} - 9y'_{k-3})$$

$$z_{k+1} = z_k + (h/24)(55z'_k - 59z'_{k-1} + 37z'_{k-2} - 9z'_{k-3})$$

unde $y'_k = f(x_k, y_k, z_k)$ și $z'_k = g(x_k, y_k, z_k)$

Formulele de prezicere furnizează seturi de valori y_k, z_k pentru folosirea formulelor corectoare de forma :

$$y_{k+1} = y_k + (h/24)(9y'_{k+1} + 19y'_k - 54y'_k + y'_{k+2})$$

$$z_{k+1} = z_k + (h/24)(9z'_{k+1} + 19z'_k - 5z'_k + z'_{k+2})$$

unde $y'_k = f(x_k, y_k, z_k)$, $z'_k = g(x_k, y_k, z_k)$ cu argumentele x_k, y_k, z_k obținute în urma folosirii formulelor de prezicere.

Ca și în cazul ecuațiilor, valorile corectate pot fi folosite din nou în partea dreaptă a formulelor de corecție pentru a obține în partea stîngă a semnelui egal o nouă corecție. Procesul se repetă pînă cînd diferența dintre două corecții succesive devine mai mică decît o valoare dată.

16. INTEGRAREA NUMERICĂ A ECUAȚIILOR DIFERENȚIALE DE ORDIN SUPERIOR

16.1. Integrarea numerică a ecuațiilor diferențiale de ordin superior cu condiții inițiale

Există o varietate de proceduri pentru integrarea acestor ecuații. O primă posibilitate constă în înlocuirea ecuației diferențiale cu condiții inițiale printr-un sistem de ecuații diferențiale de ordinul întâi cu condiții inițiale. Se rezolvă apoi sistemul prin una din metodele puse în evidență în capitolul 15. Se ilustrează această procedură pentru problema cu condiții inițiale din aplicația următoare :

Aplicația 1

Să se integreze ecuația $y'' = y$ cu condițiile inițiale $y(0) = 0$, $y'(0) = 1$. Notînd $y' = z$, se obține sistemul :

$$y' = z, \quad z' = y$$

cu condițiile $y(0) = 0$, $z(0) = 1$, sistem rezolvat prin metoda Runge-Kutta în aplicația 1 din capitolul 15.

Altă posibilitate de rezolvare a unei ecuații diferențiale de ordin superior cu condiții inițiale constă în a încerca să se determine soluția de forma unei serii de puteri. Se urmărește această procedură în aplicația următoare :

Aplicația 2

Fiind dată ecuația $y'' = y$ cu condițiile inițiale $y(0) = 0$, $y'(0) = 1$, să se determine soluția ca suma unei serii de puteri ale lui x definită în vecinătatea punctului $x = 0$, apoi să se folosească această soluție pentru calculul valorii sale în punctul $x = 0,1$.

Se caută deci o soluție $y(x)$ de forma sumei unei serii de puteri

$$y(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n + \dots$$

Considerînd că seria de mai sus poate fi derivată termen cu termen, avem :

$$y'(x) = a_1 + 2a_2x + 3a_3x^2 + \dots + na_nx^{n-1} + \dots$$

$$y''(x) = 1 \cdot 2a_2 + 2 \cdot 3a_3x + \dots + (n-1)na_nx^{n-2} + \dots$$

Egalînd y'' cu y , se obține sistemul de ecuații :

$$1 \cdot 2a_2 = a_0$$

$$2 \cdot 3a_3 = a_1$$

$$3 \cdot 4a_4 = a_2$$

$$\dots \dots \dots$$

$$(n-1)na_n = a_{n-2}$$

Dar folosind condițiile inițiale se obține : $a_0 = 0$, $a_1 = 1$. În continuare folosind sistemul de ecuații se obțin valorile coeficienților a de indice mai mari ca 1 :

$a_2 = 0$, $a_3 = 0,16666$, $a_4 = 0$, $a_5 = 0,0083$, $a_6 = 0$, ...
Rezultă o soluție de forma :

$$y(x) \simeq x + 0,1666x^3 + 0,0083x^5$$

Pentru $x = 0,1$ se obține : $y(0,1) \simeq 0,100167$

Ecuația admite soluția exactă $y = 1/2e^x - 1/2e^{-x}$ și în punctul $0,1$ valoarea exactă este $y = 0,1001667$, deci coincide cu valoarea obținută pînă la a șaptea zecimală. Pentru alte valori ale argumentului x se pot lua în considerare dezvoltări în serie în vecinătatea punctului respectiv. Adeseori metoda dezvoltării în serie servește pentru a genera valori de start pentru aplicarea altor metode.

Aplicația 3

Să se obțină o soluție de forma sumei unei serii de puteri, pentru valori ale lui x situate în vecinătatea punctului $x = 0$ pentru ecuația lui Bessel $x^2 y'' + xy' + (x^2 - \nu^2)y = 0$, $\nu \in \mathbb{R}$.

Deoarece alegerea lui $y(x)$ de forma $\sum_{x=0}^{\infty} a_i x^i$ nu conduce la nici un rezultat, vom lua :

$$y = x^{\lambda} \sum_{i=0}^{\infty} a_i x^i = \sum_{i=0}^{\infty} a_i x^{(\lambda+i)}$$

Presupunând că seria de mai sus poate fi derivată termen cu termen, se obține :

$$xy' = \sum_{i=0}^{\infty} a_i (\lambda + i) x^{\lambda+i}, \quad x^2 y'' = \sum_{i=0}^{\infty} a_i (\lambda + i)(\lambda + i - 1) x^{\lambda+i}$$

Avem : $x^2 y = \sum_{k=0}^{\infty} a_k x^{\lambda+k+2}$, sau, folosind o schimbare de indici,

$$x^2 y = \sum_{i=2}^{\infty} a_{i-2} x^{\lambda+i}$$

Introducând valorile astfel obținute în ecuația lui Bessel, ordonând după puterile lui x și impunând ca toți coeficienții lui x să fie egali cu zero, se obține sistemul de ecuații :

$$\begin{aligned} a_0(\lambda^2 - \nu^2) &= 0 \\ a_1(\lambda^2 - \nu^2 + 2\lambda + 1) &= 0 \\ \dots \dots \dots \\ a_i[(\lambda + i)^2 - \nu^2] + a_{i-2} &= 0 \end{aligned}$$

Din prima ecuație îl determinăm pe λ alegând $a_0 \neq 0$. Rezultă $\lambda = \pm \nu$, deci vom avea două soluții. Vom lua mai întâi $\lambda = +\nu$. Ecuația a doua a sistemului devine $a_1(2\nu + 1) = 0$. Dacă presupunem $\nu > 0$, rezultă $a_1 = 0$. Din sistem rezultă de asemenea că toți coeficienții de indice impar sînt zero, soluția sistemului fiind :

$$\begin{aligned} a_2 &= (-a_0)/2(2\nu + 2) \\ a_4 &= (-a_2)/4(2\nu + 4) \\ a_6 &= (-a_4)/6(2\nu + 6) \end{aligned}$$

Rezultă în final coeficienții de forma :

$$a_{0k} = (-1)^k a_0 / [2^{2k} k! (\nu + 1)(\nu + 2) \dots (\nu + k)],$$

$$k = 1, 2, \dots$$

Pentru ν întreg, alegînd pe $a_0 = 1/2^\nu \nu!$ se obține soluția ecuației Bessel notată $J_\nu(x)$ și numită *funcția Bessel de speța întâi și ordinul ν* .

$$J_\nu(x) = \frac{1}{2^\nu \nu!} x^\nu \left[1 - \frac{x^2}{4(\nu + 1)} + \frac{x^4}{32(\nu + 1)(\nu + 2)} - \frac{x^6}{384(\nu + 1)(\nu + 2)(\nu + 3)} + \dots \right] = \sum_{p=0}^{\infty} \frac{(-1)^p (x/2)^{\nu+2p}}{p! \Gamma(\nu + p + 1)}$$

unde $\Gamma(\nu + p + 1)$ este funcția Gamma. Deoarece $\nu + p + 1$ reprezintă o valoare întreagă, avem: $\Gamma(\nu + p + 1) = (\nu + p)!$ Soluția ecuației Bessel a fost obținută deci ca suma unei serii de puteri cu raza de convergență

$$R = \lim_{p \rightarrow \infty} \left| \frac{(-1)^p}{p! \Gamma(\nu + p + 1)} \cdot \frac{(p + 1)! \Gamma(\nu + p + 2)}{(-1)^{p+1}} \right| = \infty$$

deci seria converge pe toată axa reală.

Pentru $\nu = 0$, soluția ecuației Bessel devine :

$$J_0(x) = 1 - \frac{x^2}{2^2} + \frac{x^4}{2^2 4^2} - \frac{x^6}{2^2 4^2 6^2} + \dots$$

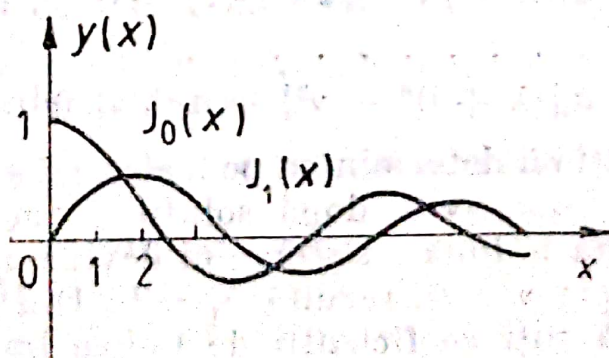


Fig. 16.1

avînd graficul dat în fig. 16.1

O funcție $J_{-\nu}^{(x)}$, $\nu > 0$ poate fi definită înlocuind în relația de definiție a lui $J_\nu^{(x)}$ pe ν prin $-\nu$. Dacă ν este întreg, se

poate arăta că $J_{\nu}^{(x)} = (-1)^n J_{\nu}^{(x)}$. Dacă ν nu este întreg, $J_{\nu}^{(x)}$ și $J_{-\nu}^{(x)}$ sînt liniar independente, deci în acest caz soluția generală a ecuației Bessel va fi :

$$y = c_1 J_{\nu}(x) + c_2 J_{-\nu}(x), \quad \nu \notin \mathbb{N}$$

16.2. Integrarea numerică a ecuațiilor diferențiale de ordin superior cu condiții la limită

Dacă ecuația diferențială este liniară, se poate înlocui problema la limită prin probleme cu condiții inițiale. Fie de exemplu ecuația de ordinul doi $y'' = p(x)y$ cu condițiile la limită $y(x_0) = y_0$, $y(x_1) = y_1$. Se începe prin rezolvarea numerică conform metodelor din cap. 14 a două probleme cu condiții inițiale :

$$\begin{aligned} y_1'' &= p(x)y_1, & y_1(x_0) &= 1, & y_1'(x_0) &= 0 \\ y_2'' &= p(x)y_2, & y_2(x_0) &= 0, & y_2'(x_0) &= 1 \end{aligned}$$

Folosind principiul superpozițiilor pentru ecuația liniară, se caută soluția $y(x)$ de forma $y(x) = c_1 y_1(x) + c_2 y_2(x)$, unde constantele c_1 și c_2 sînt determinate folosind condițiile la limită. Se ilustrează această procedură pentru problema la limită din aplicația următoare :

Aplicația 4

Să se integreze ecuația diferențială liniară de ordinul doi $y'' = y$ cu condițiile la limită $y(0) = 0$, $y(1) = 1$. Conform celor arătate, se începe prin rezolvarea a două probleme cu condiții inițiale :

$$\begin{aligned} y_1'' &= y_1, & y_1(0) &= 1, & y_1'(0) &= 0 \\ y_2'' &= y_2, & y_2(0) &= 0, & y_2'(0) &= 1 \end{aligned}$$

Se pot utiliza metodele de integrare numerică descrise în cap. 14 (de exemplu Runge-Kutta) continuate prin formarea polinoamelor de aproximare, dar, în acest caz, pentru ușurință, se folosește soluția exactă a ecuației $y(x) = k_1 e^x +$

$+ k_2 e^{-x}$. Folosind condițiile la limită pentru prima problemă inițială se obține sistemul de ecuații $k_1 + k_2 = 1, k_1 - k_2 = 0$, cu soluția $k_1 = k_2 = 1/2$.

Rezultă soluția primei probleme cu condiții inițiale de forma :

$$y_1(x) = \frac{1}{2} (e^x + e^{-x})$$

Procedînd asemănător pentru a doua problemă cu condiții inițiale, se obține soluția $y_2(x) = \frac{1}{2} (e^x - e^{-x})$

Se caută în continuare conform metodei expuse soluția problemei la limită de forma $y = c_1 y_1(x) + c_2 y_2(x)$, deci $y = c_1 (e^x + e^{-x}) + c_2 (e^x - e^{-x})$.

Folosind sistemul de condiții la limită, se obține sistemul de ecuații :

$$2c_1 = 0$$

$$c_1(e + 1/e) + c_2(e - 1/e) = 1$$

cu soluția :

$$c_1 = 0, c_2 = e/(e^2 - 1)$$

Rezultă deci soluția problemei la limită de forma :

$$y(x) = \frac{e}{e^2 - 1} (e^x - e^{-x})$$

Altă metodă care poate fi folosită și pentru integrarea ecuațiilor neliniare cu condiții la limită se exprimă astfel : fie ecuația $y'' = f(x, y, y')$ cu condițiile la limită $y(x_0) = y_0, y(x_1) = y_1$. Se începe prin înlocuirea acestei probleme printr-o problemă cu condiții inițiale folosind un parametru α :

$$y'' = f(x, y, y'), y(x_0) = y_0, y'(x_0) = \alpha$$

Se rezolvă problema cu condiții inițiale astfel obținută și fie $y(x, \alpha)$ soluția sa. Trebuie determinat parametrul α astfel ca să fie îndeplinită și a doua condiție la limită, adică $y(x_1, \alpha) = y_1$. Cu alte cuvinte, trebuie rezolvată o ecuație algebrică cu necunoscuta α .

Aplicația 5

Să se integreze ecuația diferențială de ordinul doi din aplicația 4, $y'' = y$ cu condițiile la limită $y(0) = 0$, $y(1) = 1$. Procedând conform celor prezentate anterior, se începe prin integrarea ecuației cu condiții inițiale: $y'' = y$, $y(0) = 0$, $y'(0) = \alpha$.

Se obține soluția $y(x) = (\alpha/2)(e^x + e^{-x})$. Se determină parametrul α folosind a doua condiție la limită: $y(1) = \frac{\alpha}{2} \left(e + \frac{1}{e} \right) = 1$.

Rezultă $\alpha = 2e/(e^2 + 1)$. În final se regăsește soluția ecuației cu condiții la limită

$$y(x) = \frac{e}{e^2 + 1} (e^x + e^{-x}).$$

Altă metodă pentru integrarea unei ecuații diferențiale cu condiții la limită constă în înlocuirea ecuației diferențiale printr-o ecuație cu diferențe finite. Se ilustrează în continuare această metodă pentru ecuația tratată și în exemplele precedente.

Aplicația 6

Să se integreze ecuația diferențială de ordinul doi $y'' = y$ cu condițiile la limită $y(0) = 0$, $y(1) = 1$.

Se efectuează mai întâi o diviziune a intervalului $[0, 1]$ în n subintervale egale de lungime h folosind punctele intermediare

$$0 = x_0, x_1, x_2, \dots, x_n = 1$$

Se aproximează y'' folosind diferențele finite:

$$y'' \simeq \frac{\Delta^2 y_i}{h^2} = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}$$

Înlocuind derivata a doua astfel obținută în ecuația dată și folosind condițiile la limită, se obține sistemul de $(n + 1)$ ecuații cu $(n + 1)$ necunoscute $y_0, y_1, y_2, \dots, y_n$.

$$y_{i+1} - y_i(2 + h^2) + y_{i-1} = 0; i = 1, 2, \dots, (n - 1)$$

$$y_0 = 0$$

$$y_n = 1$$

De exemplu, lucrînd cu diviziunea

$$0 = x_0, x_1 = 1/2, x_2 = 1$$

și avînd pasul $h = 1/2$, se obține soluția aproximativă a problemei la limită sub forma tabelului de valori

x_k	0	0,5	1
y_k	0	0,444	1

17. REZOLVAREA NUMERICĂ A ECUAȚIILOR DIFERENȚIALE CU DERIVATE PARȚIALE

17.1. Ecuații de tip eliptic

Se ilustrează metoda de calcul pentru ecuația lui Laplace :

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, \quad 0 \leq x \leq 1, \quad 0 \leq y \leq 1$$

cu condiții la limită :

$$u(0, y) = u(1, y) = u(x, 1) = 0$$

$$u(x, 0) = 1$$

altfel spus, rezolvând problema lui Dirichlet. Se începe prin acoperirea domeniului pe care este definită funcția $u(x, y)$, cu o rețea de elemente pătrate sau dreptunghiulare, din care cauză metoda de rezolvare este denumită *metoda rețelelor*. În fig. 17.1 s-a ales o rețea de elemente pătrate care determină 4 noduri interioare, deci se lucrează cu o rețea având pasul h (latura pătratului) egală cu 0,33. Se aproximează derivatele parțiale de ordinul al doilea cu ajutorul diferențelor finite :

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u(x+h, y) - 2u(x, y) + u(x-h, y)}{h^2}$$

$$\frac{\partial^2 u}{\partial y^2} \approx \frac{u(x, y+h) - 2u(x, y) + u(x, y-h)}{h^2}$$

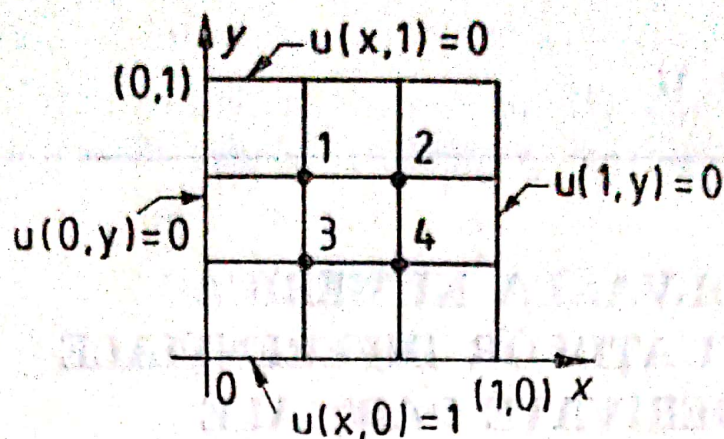


Fig. 17.1

Ecuatia lui Laplace devine :

$$\frac{u(x+h, y) - 2u(x, y) + u(x-h, y)}{h^2} + \frac{u(x, y+h) - 2u(x, y) + u(x, y-h)}{h^2} = 0$$

de unde rezultă :

$$u(x, y) = \frac{1}{4} [u(x+h, y) + u(x-h, y) + u(x, y+h) + u(x, y-h)]$$

Ultima relație pune în evidență că valoarea funcției u în unul din punctele interioare se calculează ca media aritmetică a valorilor lui u în cele patru noduri vecine (noduri de asemenea interioare sau situate pe frontieră). Pentru a simplifica notația, să notăm valoarea lui u în nodul 1 prin u_1 , în nodul 2 prin u_2 , etc. Se obține astfel, folosind și condițiile la limită, sistemul de ecuații :

$$\begin{aligned} x_1 &= 1/4(0 + x_2 + x_3 + 0) & x_3 &= 1/4(0 + x_1 + x_4 + 1) \\ x_2 &= 1/4(x_1 + 0 + 0 + x_4) & x_4 &= 1/4(x_2 + x_3 + 0 + 1) \end{aligned}$$

Forma sistemului de ecuații permite utilizarea metodei iterative pentru rezolvarea sa. Folosind un set de valori inițiale, sistemul de ecuații a fost rezolvat în aplicația 3, cap.3. Rezultă că prin metoda rețelei, pornind de la ecuația lui Laplace și de la valorile funcției u la periferia domeniului

de definiție (un pătrat, în acest caz), se pot calcula aproximativ valorile funcției în cele patru puncte interioare.

Pentru a evalua ordinul de mărime al erorii care se comite folosind acest algoritm de calcul, se dezvoltă după formula lui Taylor funcțiile care intervin în algoritm, limitându-ne la primii 3 termeni :

$$u(x - h, y) = u - \frac{h}{1!} u'_x + \frac{h^2}{2!} \ddot{u}_{x^2}$$

$$u(x + h, y) = u + \frac{h}{1!} u'_x + \frac{h^2}{2!} \ddot{u}_{x^2}$$

$$u(x, y - h) = u - \frac{h}{1!} u'_y + \frac{h^2}{2!} \ddot{u}_{y^2}$$

$$u(x, y + h) = u + \frac{h}{1!} u'_y + \frac{h^2}{2!} \ddot{u}_{y^2}$$

Rezultă :

$$u(x, y) - \frac{1}{4} [u(x + h, y) + u(x - h, y) + u(x, y + h) + u(x, y - h)] = -\frac{h^2}{4} (\ddot{u}_{x^2} + \ddot{u}_{y^2})$$

Se observă că eroarea este proporțională cu pătratul lui h . Micșorînd valoarea pasului h , deci lucrînd cu o rețea mai fină, se micșorează și eroarea, dar în același timp crește numărul ecuațiilor din sistem.

De obicei problema se rezolvă la început pentru valori mai mari ale lui h , adică pentru un număr mai mic de noduri interioare și numai după ce problema a fost rezolvată aproximativ pentru această rețea rară, se trece la o rețea mai fină pentru întreg domeniul sau numai pentru o parte a sa.

Algoritmul de calcul conduce la un sistem liniar neomogen în care numărul ecuațiilor este egal cu numărul necunoscutelor. Se poate arăta că acest sistem admite întotdeauna o soluție unică. Pentru a dovedi să rescriem sistemul sub forma :

$$\sum_{k=1}^n a_{ik} u_k = b_i, \quad i = 1, 2, \dots, n$$

Termenii liberi b_i din aceste ecuații depind de condițiile la limită date. Dacă toate aceste valori sînt egale cu zero, rezultă $b_i = 0$, iar sistemul devine omogen.

$$\sum_{k=1}^n a_{ik} u_k = 0, \quad i = 1, 2, \dots, n$$

Conform unei teoreme fundamentale de algebră, sistemul de ecuații neomogene va avea o soluție unică dacă sistemul de ecuații omogene va avea numai soluția nulă. Să presupunem că sistemul de ecuații omogene ar avea și soluții strict pozitive. Atunci și cea mai mare valoare a acestora va fi de asemenea strict pozitivă. Să presupunem că valoarea maximă este atinsă în punctul interior x . Datorită algoritmului de calcul de forma :

$$u_x = 1/4(u_A + u_B + u_C + u_D)$$

rezultă că valoarea maximă trebuie să fie atinsă și în unele din punctele vecine A, B, C și D . Într-adevăr, conform presupunerii inițiale, avem :

$$u_A \leq u_x, \quad u_C \leq u_x$$

$$u_B \leq u_x, \quad u_D \leq u_x$$

Dar nici una dintre aceste inegalități nu poate fi strictă, deoarece în acest caz, utilizînd algoritmul de calcul, ar rezulta : $u_x < u_x$, ceea ce este absurd. Continuînd acest raționament, rezultă, mergînd din punct în punct, că putem transfera valoarea maximă pe frontiera domeniului, dar acolo toate valorile lui u sînt egale cu zero (pentru a se obține sistemul de ecuații omogen), ceea ce contrazice faptul că valoarea maximă este strict pozitivă. Rezultă atunci că presupunerea făcută, și anume că sistemul de ecuații omogene ar avea și soluții strict pozitive, este falsă. În același fel se arată că sistemul nu poate avea nici soluții strict negative. Rezultă deci că sistemul de ecuații omogene are toate soluțiile egale cu zero.

În practică se întîmplă frecvent ca domeniul pe care este definită funcția necunoscută u să aibă o frontieră curbilinie (fig. 17.2)

Determinarea funcției u în punctul I , conform algoritmului de calcul, se face astfel :

$$u_I = \frac{1}{4}(u_A + u_B + u_C + u_D)$$

deci presupune cunoașterea funcției u în punctul B situat în afara domeniului.

Putem calcula funcția u în punctul B cunoscând valoarea lui u în punctul F situat pe frontieră, folosind o formulă de interpolare liniară :

$$\frac{u(B) - u(F)}{\delta} = \frac{u(F) - u(I)}{h - \delta} \text{ sau } u(B) = u(F) + \frac{\delta}{h - \delta} [u(F) - u(I)] ; \delta < h$$

Dacă $\delta = 0$, rezultă $u_B = u_F$. Ecuația de mai sus se adaugă sistemului de ecuații la care conduce metoda rețelei, mărind și numărul necunoscutelor cu o unitate, și anume cu u_B .

17.2. Ecuații de tip parabolic

Se ilustrează metoda pentru ecuația căldurii într-o bară de lungime l :

$$\frac{1}{a^2} \frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = 0,$$

unde $u(x, t)$ reprezintă temperatura ce depinde de coordonata x a punctului și de timp. Dacă se face schimbarea de variabilă $\bar{t} = a^2 t$, rezultă că

$$\frac{\partial u}{\partial \bar{t}} - \frac{\partial^2 u}{\partial x^2} = 0$$

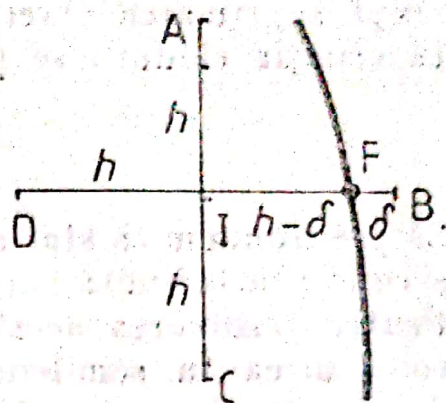


Fig. 17.2

După suprimarea barei din expresia noii variabile t , rezultă că ecuația căldurii se poate scrie :

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = 0.$$

Să presupunem că sînt date condiția inițială : $u(x, 0) = f(x)$ și condițiile la limită : $u(0, t) = \alpha(t)$, $u(l, t) = \beta(t)$.

Pentru rezolvarea acestei probleme, metoda rețelei impune construirea în semibanda $t \geq 0$, $0 \leq x \leq l$ a unei rețele dreptunghiulare (fig. 17.3).

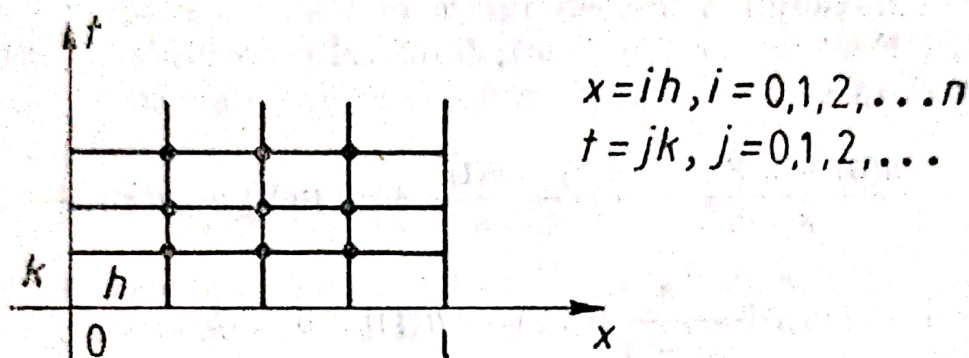


Fig. 17.3

Se alege pasul $h = l/n$ iar $k = \sigma h^2$ (σ reprezintă o constantă ce urmează să fie determinată). Aproximînd derivatele parțiale prin diferențe finite, ecuația se poate scrie în nodurile rețelei notate cu doi indici, astfel :

$$\frac{u_{i,j+1} - u_{i,j}}{\sigma h^2} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2}$$

sau $u_{i,j+1} = \sigma u_{i-1,j} + (1 - 2\sigma)u_{i,j} + u_{i+1,j}$

Se observă că pentru calcule se folosesc patru puncte vecine (fig. 17.4)

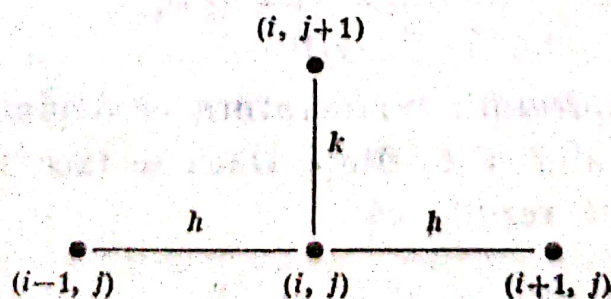


Fig. 17.4



În punctele situate pe axa Ox (deci pentru $t = 0$) valorile funcției u rezultă din condiția inițială $u(x_i, 0) = f(x_i)$, $i = 0, 1, 2, \dots, n$. Condițiile la limită conduc la :

$$u(0, t_j) = \alpha(t_j)$$

$$u(l, t_j) = \beta(t_j), j = 0, 1, \dots$$

Algoritmul de calcul permite ca folosind condițiile inițiale și condițiile la limită, să determinăm valorile funcției căutate $u(x, t)$ în toate nodurile semibenzii alese.

Se determină valoarea constantei σ din condiția ca eroarea ce apare datorită înlocuirii ecuației cu derivate parțiale cu ecuația cu diferențe finite, să fie minimă. Procedând asemănător ca în cazul rezolvării problemei Dirichlet, se apreciază că eroarea este minimă pentru $\sigma = 1/6$.

Rezultă că algoritmul de calcul se poate scrie :

$$u_{i,j} = \frac{1}{6} (u_{i-1,j} + 4u_{i,j} + u_{i+1,j})$$

$$u_{i,0} = f_i, u_{0,j} = \alpha_j, u_{n,j} = \beta_j$$

unde s-a notat : $f_i = f(x_i)$, $\alpha_j = \alpha(t_j)$, $\beta_j = \beta(t_j)$

iar $n = l/h$ reprezintă numărul de subdiviziuni în direcția x .

Ecuația cu diferențe finite este liniară cu coeficienți constanți și poate fi rezolvată prin metoda separării variabilelor. Punînd $u_{i,j} = X_i T_j$, ecuația devine :

$$\frac{X_{i-1} + 4X_i + X_{i+1}}{6X_i} = \frac{T_{j+1}}{T_j} = -\lambda$$

În ecuația de mai sus λ este o constantă deoarece fiecare membru al ecuației este funcție de cîte o singură variabilă independentă x sau t . Folosind separarea variabilelor, ecuația poate fi înlocuită de sistemul cu două ecuații cu diferențe finite :

$$X_{i+1} + (4 + 6\lambda)X_i + X_{i-1} = 0$$

$$T_{j+1} + \lambda T_j = 0$$

care poate fi rezolvat conform metodelor date în cap. 18.

17.3. Ecuații de tip hiperbolic

Se ilustrează metoda pentru ecuația oscilațiilor libere ale unei corzi omogene finite: $\frac{1}{a^2} \frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} = 0$ însoțită de condițiile inițiale: $u(x, 0) = f(x)$, $u_t'(x, 0) = g(x)$; $0 \leq x \leq l$ și condițiile la limită:

$$u(0, t) = \alpha(t), \quad u(l, t) = \beta(t); \quad 0 \leq t < \infty$$

Se procedează analog cazului precedent (ecuația căldurii), acoperind semibanda $0 < x \leq l$, $0 \leq t < \infty$ cu o rețea dreptunghiulară, alegînd pe $h = l/n$ și k întreg oarecare. Înlocuind derivatele parțiale prin diferențe finite, ecuația se poate scrie în nodurile rețelei:

$$\frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{k^2} = a^2 \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2}$$

Pentru $h = ak$ ecuația poate fi adusă la o formă mai simplă:

$$u_{i,j+1} = u_{i+1,j} + u_{i,j-1} + u_{i-1,j}$$

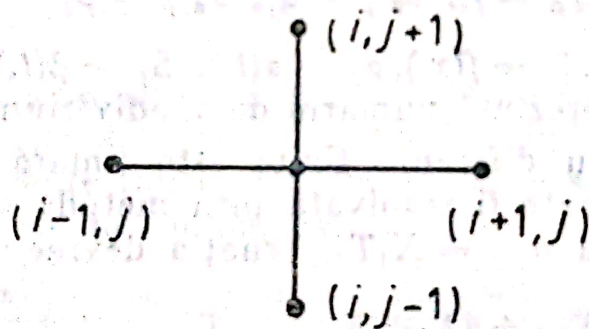


Fig. 17.4

Se observă din fig. 17.4 că algoritmul de calcul permite calculul valorii funcției u în punctul $(i, j+1)$ dacă sînt cunoscute valorile lui u în celelalte 3 puncte. Dar prima condiție inițială nu este suficientă pentru pornirea calculelor. Din această cauză se folosește a doua condiție inițială, aproximînd anterior derivata cu diferențe finite. Atunci:

$$\frac{u_{i,-1} - u_{i,0}}{-k} = g_i, \quad \text{unde s-a notat } g_i = g(x_i). \quad \text{Rezultă:}$$

$u_{i,-1} = u_{i,0} - kg_i$. Cu această ultimă relație calculele pot fi începute pe baza algoritmului de calcul dat. Se folosesc condițiile la limită pentru determinarea valorilor: u_{0j} și u_{nj} .

18. ECUAȚII RECURENTE CU DIFERENȚE FINITE

18.1. Introducere

Diferențele finite sînt folosite pentru a transforma ecuațiile diferențiale sau cu derivate parțiale în relații de recurență între diferite valori y_k ale unei funcții definite pe o mulțime discretă de argumente x_k . Fiind dată de exemplu problema cu valori inițiale: $y' = y$, $y(0) = 1$, aproximînd derivata cu ajutorul diferențelor finite, se poate înlocui ecuația diferențială prin ecuația recurentă $y_{k+1} = (h+1)y_k$, $y_0 = 1$. O soluție a ecuației cu diferențe finite este o secvență de valori y_k pentru care ecuația este verificată pentru un set de valori întregi k . În cazul ecuației de mai sus se verifică imediat că soluția este $y_k = (h+1)^k$, $k = 1, 2, \dots$. Ordinul unei ecuații recurente este diferența dintre cel mai mare și cel mai mic indice care intervine în expresia ecuației. Se pot stabili strînse analogii între teoria ecuațiilor diferențiale și teoria ecuațiilor recurente cu diferențe finite.

În cazul ecuației diferențiale anterioare, datorită condiției inițiale, soluția este unică și poate fi obținută printr-o integrare de forma $y = e^x$. Ecuația cu diferențe finite care îi corespunde are de asemenea o soluție unică, obținută prin însumare: $y_k = (h+1)^k \simeq e^{hk}$. Soluția exactă a ecuației diferențiale pentru $x_k = kh$ dă valoarea: $y(x_k) = e^{x_k}$.

Dacă în mod obișnuit soluția generală a ecuației recurente însoțită de condiție inițială se obține sub forma unei secvențe numerice, în unele cazuri ea poate fi exprimată analitic în funcție de argumentul k .

Aplicația 1

Fie ecuația : $y_{k+1} = ky_k + 1$, cu $y_0 = 1$.

Avem :

$$y_1 = k + 1$$

$$y_2 = k(k + 1) = k^2 + k + 1$$

...

$$y_n = k^n + k^{n-1} + \dots + 1 = \frac{1 - k^n}{1 - k}$$

Se prezintă în continuare modul de rezolvare a ecuației recurente liniare de ordinul doi care intervine frecvent în rezolvarea numerică a ecuațiilor diferențiale sau cu derivate parțiale.

18.2. Ecuația omogenă de ordinul doi

Fie ecuația : $y_{k+2} + ay_{k+1} + by_k = 0$ în care coeficienții a și b depind numai de k . Să presupunem că u_k și v_k sînt două soluții ale ecuației omogene, adică :

$$u_{k+2} + au_{k+1} + bu_k = 0, \quad v_{k+2} + av_{k+1} + bv_k = 0.$$

Înmulțind relațiile de mai sus cu două constante arbitrare c_1 , respectiv c_2 și adunînd, obținem : $c_1 u_{k+2} + c_2 v_{k+2} + a(c_1 u_{k+1} + c_2 v_{k+1}) + b(c_1 u_k + c_2 v_k) = 0$, fapt ce demonstrează că $c_1 u_k + c_2 v_k$ reprezintă de asemenea o soluție. Această proprietate este asemănătoare cu principiul superpozițiilor pentru ecuațiile diferențiale liniare. Se poate arăta că dacă determinantul denumit Wronskian, $w_k = \begin{vmatrix} u_k & v_k \\ u_{k-1} & v_{k-1} \end{vmatrix} \neq 0$, atunci orice soluție a ecuației poate fi reprezentată ca o combinație liniară cu cele două soluții particulare u_k și v_k , $y_k = c_1 u_k + c_2 v_k$.

Această proprietate are de asemenea un corespondent în teoria generală a ecuațiilor diferențiale liniare. Dacă coeficienții ecuației a și b sînt constanți, soluțiile u_k și v_k pot fi determinate cu ușurință în modul următor: se formează ecuația caracteristică: $r^2 + ar + b = 0$ și fie r_1 și r_2 rădăcinile acestei ecuații. În funcție de natura acestor rădăcini, avem următoarele soluții:

- 1) $a^2 - 4b > 0$, adică $r_1 \neq r_2$, rezultă $u_k = r_1^k$, $v_k = r_2^k$;
- 2) $a^2 - 4b = 0$, adică $r_1 = r_2$, rezultă $u_k = r^k$, $v_k = kr^k$;
- 3) $a^2 - 4b < 0$, adică rădăcini complex conjugate, atunci notînd forma trigonometrică a rădăcinilor $r_{1,2} = \rho(\cos \varphi \pm \pm i \sin \varphi)$, rezultă: $u_k = \rho^k \sin k\varphi$, $v_k = \rho^k \cos k\varphi$.
Cele de mai sus pot fi ușor verificate. Într-adevăr, să luăm cazul rădăcinilor reale și distincte ale ecuației caracteristice. Evident $u_k = r_1^k$ este soluția ecuației recurente, deoarece:

$$u_{k+2} + au_{k+1} + bu_k = r_1^k(r_1^2 + ar_1 + b) = 0.$$

Aplicația 2

Să se rezolve ecuația liniară și omogenă de ordinul 2:

$$y_{k+2} = y_{k+1} + y_k, \quad y_0 = 0, \quad y_1 = 1$$

Pornind de la valorile inițiale și dînd lui k valori din ce în ce mai mari, se obține o secvență cunoscută sub numele de numerele lui Fibonacci:

1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

Se poate utiliza teoria generală pentru a obține soluția problemei sub formă analitică. Astfel ecuația caracteristică $r^2 - r - 1 = 0$ are două rădăcini reale și distincte: $r_{1,2} = (1 \pm \sqrt{5})/2$. Rezultă soluția ecuației recurente de forma:

$$y_k = c_1 u_k + c_2 v_k = c_1 \left(\frac{1 + \sqrt{5}}{2} \right)^k + c_2 \left(\frac{1 - \sqrt{5}}{2} \right)^k$$

Folosind condițiile inițiale, avem:

$$c_1 + c_2 = 0$$

$$c_1 \left(\frac{1 + \sqrt{5}}{2} \right) + c_2 \left(\frac{1 - \sqrt{5}}{2} \right) = 1$$

Rezultă: $c_1 = -c_2 = 1/\sqrt{5}$.

În final :

$$y_k = \frac{1}{\sqrt{5}} \left[\left(\frac{1 + \sqrt{5}}{2} \right)^k - \left(\frac{1 - \sqrt{5}}{2} \right)^k \right]$$

Numerele lui Fibonacci intervin în anumite probleme ce implică transferul informațiilor de-a lungul unui canal de comunicații.

18.3. Ecuația neomogenă de ordinul doi

Fie ecuația : $y_{k+2} + ay_{k+1} + by_k = \alpha_k$
 Asemănător cu teoria ecuațiilor diferențiale liniare neomogene, se poate arăta și în acest caz că dacă u_k și v_k sînt soluții ale ecuației omogene cu wronskianul nenul și Y_k este o soluție particulară a ecuației neomogene, atunci orice soluție a ecuației poate fi scrisă : $y_k = c_1 u_k + c_2 v_k + Y_k$, c_1 și c_2 fiind constante.

Într-adevăr, scriind că y_k și Y_k sînt soluții pentru ecuația neomogenă, avem :

$$y_{k+2} + ay_{k+1} + by_k = \alpha_k$$

$$Y_{k+2} + aY_{k+1} + bY_k = \alpha_k$$

Notînd : $y_k - Y_k = d_k$ și scăzînd cele două relații de mai sus, obținem : $d_{k+2} + ad_{k+1} + bd_k = 0$.

Acest fapt ce arată că d_k reprezintă o soluție a ecuației omogene, deci

$$d_k = c_1 u_k + c_2 v_k.$$

În final rezultă : $y_k = c_1 u_k + c_2 v_k + Y_k$.

Prin analogie cu teoria ecuațiilor diferențiale liniare și neomogene, pentru anumite forme ale termenului liber α_k se pot încerca soluții particulare Y_k de tipul următor :

$$\alpha_k = Ax^k \rightarrow Y_k = Cx^k$$

$$\alpha_k = k^n \rightarrow Y_k = C_0 + C_1 k + C_2 k^2 + \dots + C_n k^n$$

$$\alpha_k = k^n x^k \rightarrow Y_k = x^k (C_0 + C_1 k + C_2 k^2 + \dots + C_n k^n)$$

$\alpha_k = \sin Ak$ sau $\alpha_k = \cos Ak \rightarrow Y_k = C_1 \sin Ak + C_2 \cos Ak$
 $\alpha_k = x^k \sin Ak$ sau $\alpha_k = x^k \cos Ak \rightarrow Y_k = x^k (C_1 \sin Ak + C_2 \cos Ak)$.

Aplicația 3

Să se rezolve ecuația neomogenă : $y_{k+2} - y_{k+1} - y_k = x^k$
 Se încearcă o soluție particulară pentru ecuația neomogenă de forma $Y_k = Cx^k$. Avem : $Cx^k(x^2 - x + 1) = x^k$. Rezultă : $C = 1/(x^2 - x - 1)$. Cum ecuația omogenă a fost rezolvată în exemplul 2, soluția generală este :

$$y_k = c_1 \left(\frac{1 + \sqrt{5}}{2} \right)^k + c_2 \left(\frac{1 - \sqrt{5}}{2} \right)^k + \frac{x^k}{x^2 - x - 1}$$

19. REZOLVAREA NUMERICĂ A ECUAȚILOR INTEGRALE

19.1. Introducere

O ecuație se numește *integrală* dacă funcția necunoscută căutată se găsește sub semnul de integrală. De exemplu ecuația : $f(x) - f(a) = \int_a^x y(t)dt$, unde $f(x)$ este o funcție dată definită pe intervalul $[a, b]$, iar funcția $y(t)$ este funcția necunoscută, are evident soluția $y(t) = \frac{df(t)}{dt}$. De asemenea, probleme pentru rezolvarea ecuațiilor diferențiale ordinare pot fi rearanjate ca ecuații integrale. De exemplu, problema lui Cauchy pentru ecuația $\frac{dx}{dt} = F[t, x(t)]$ impune determinarea soluției ecuației diferențiale care verifică condiția inițială $x(a) = x_0$. Fie $x = \varphi(t)$ o soluție a problemei enunțate. Înlocuind această soluție în ecuația diferențială, obținem identitatea : $\varphi'(t) = F[t, \varphi(t)]$. Integrând de la a la t , obținem : $\varphi(t) = x_0 + \int_a^t F[y, \varphi(y)] dy$, deci funcția $\varphi(t)$ soluție a ecuației diferențiale, este totodată soluție a unei ecuații integrale neliniare de tip Volterra.

Ecuațiile integrale se întâlnesc adeseori în modelarea matematică a problemelor referitoare la încovoieri sau tor-

siuni elastice, de asemenea probleme de propagare a căldurii. Ecuațiile integrale liniare se pot clasifica în funcție de tipul și speța ecuației respective :

Tipul	Speța	Forma ecuației
Fredholm	I	$f(x) = \int_a^b N(x, y)\varphi(y) dy$
	II	Forma neomogenă $\varphi(x) = f(x) + \lambda \int_a^b N(x, y)\varphi(y) dy$ Forma omogenă $\varphi(x) = \lambda \int_a^b N(x, y)\varphi(y) dy$
Volterra	I	$f(x) = \int_a^x N(x, y)\varphi(y) dy$
	II	Forma neomogenă $\varphi(x) = f(x) + \lambda \int_a^x N(x, y)\varphi(y) dy$ Forma omogenă $\varphi(x) = \lambda \int_a^x N(x, y)\varphi(y) dy$

În ecuațiile precedente φ reprezintă funcția necunoscută, f și N sînt funcții date, iar λ este un parametru numeric real. Funcția N poartă numele de *nucleul ecuației integrale* și de obicei este definită în pătratul $a \leq x \leq b$, $a \leq y \leq b$. Funcția f reprezintă termenul liber al ecuației, dacă ecuația este neomogenă și este definită pe $[a, b]$.

În ecuațiile integrale neliniare funcția necunoscută φ apare într-o formă neliniară, deci ecuația se scrie :

$$\varphi(x) = f(x) + \lambda \int_a^b h[x, y, \varphi(y)] dy$$

unde h este o funcție dată. Ecuatiile integrale neliniare sînt foarte diferite, ceea ce face clasificarea lor dificilă. Se prezintă în continuare metoda aproximațiilor succesive pentru integrarea ecuațiilor integrale liniare.

19.2. Integrarea ecuației tip Fredholm neomogenă de speța a doua prin metoda aproximațiilor succesive

Fie această ecuație :

$$\varphi(x) = f(x) + \lambda \int_a^b N(x, y) \varphi(y) dy.$$

Presupunînd că funcțiile f și N sînt continue pe domeniul lor de deliniție și că parametrul numeric λ este suficient de mic, putem lua ca soluție inițială pe $\varphi_1(x) = f(x)$. Se construiește șirul aproximațiilor succesive :

$$\varphi_k(x) = f(x) + \lambda \int_a^b N(x, y) \varphi_{k-1}(y) dy$$

Metoda are dezavantajul că impune la fiecare iterație calculul efectiv al integralei care apare în partea dreaptă a semnului egal.

Pentru a obține o condiție de convergență pentru șirul aproximațiilor succesive, scădem pe $\varphi_k(x)$ din $\varphi(x)$. Obținem :

$$\varepsilon_k(x) = \varphi(x) - \varphi_k(x) = \lambda \int_a^b N(x, y) \varepsilon_{k-1}(y) dy$$

Să presupunem că nucleul N este mărginit, deci există $M > 0$, astfel că $|N(x, y)| \leq M$ oricare ar fi punctul (x, y) din pătratul $[a, b] \times [a, b]$. Să notăm cu R_k valoarea maximă a lui $|\varepsilon_k(x)|$ din intervalul $[a, b]$ iar cu $\bar{x}$ punctul în care această valoare este atinsă. Atunci folosind formula de medie pentru integrala definită, obținem :

$$\lambda \int_a^b |N(\bar{x}, y)| |\varepsilon_{k-1}(y)| dy \leq \lambda M(b - a) R_{k-1}$$

Rezultă că șirul iterațiilor converge dacă $\lambda M(b-a) < 1$. Calculele pot fi ușurate dacă integrala care apare în partea dreaptă a semnelui egal în șirul iterațiilor se calculează numeric. Pentru aceasta se presupune că intervalul $[a, b]$ a fost divizat în n subintervale folosind puncte intermediare x_i egal distanțate cu pasul h . Dacă notăm :

$$f(x_i) = f_i, \quad \varphi(x_i) = \varphi_i, \quad N(x_i, y_j) = N_{ij},$$

atunci relația de iterare poate fi rescrisă în forma :

$$\varphi_i = f_i + \lambda \int_a^b N(x_i, y) \varphi(y) dy \simeq f_i + \lambda h \sum_{j=0}^n A_j N_{ij} \varphi_j,$$

unde ponderile A_j depind de formula de cvadratură folosită. Notînd $C_{ij} = h A_j N_{ij}$ obținem sistemul de ecuații liniare

$$\varphi_i = f_i + \lambda \sum_{j=0}^n C_{ij} \varphi_j, \quad i = 0, 1, \dots, n$$

Rezolvarea sistemului de ecuații liniare se poate face cu ușurință prin metoda iterativă, folosind de exemplu o soluție inițială $\varphi_i^{(0)} = 0$. Atunci :

$$\varphi_i^{(k)} = f_i + \lambda \sum_{j=0}^n C_{ij} \varphi_j^{(k-1)}, \quad i = 0, 1, \dots, n$$

unde $\varphi_i^{(k)}$ reprezintă valoarea funcției $\varphi(x_i)$ la iterația k .

19.3. Integrarea ecuației tip Volterra neomogenă de speța a doua prin metoda aproximațiilor succesive

Fie această ecuație :

$$\varphi(x) = f(x) + \int_a^x N(x, y) \varphi(y) dy$$

presupunînd că s-a înglobat λ în nucleul N . Folosind ecuația de mai sus se poate calcula diferența finită de ordinul întâi pentru $\varphi(x)$:

$$\varphi(x) - \varphi(x-h) = f(x) - f(x-h) + \int_{x-h}^x N(x, y) \varphi(y) dy$$

Utilizând o formulă de integrare numerică și folosind notațiile de mai sus, obținem :

$$\varphi_i - \varphi_{i-1} = f_i - f_{i-1} + \sum_{j=0}^i C_{ij} \varphi_j$$

Relația anterioară poate fi rescrisă astfel :

$$\varphi_i = \frac{1}{1 - C_{ii}} [f_i - f_{i-1} + (1 + C_{i, i-1}) \varphi_{i-1} + \sum_{j=0}^{i-2} C_{ij} \varphi_j]$$

Din ecuația Volterra neomogenă rezultă că $\varphi(a) = f(a)$, deci $\varphi_0 = f_0$. Din relația precedentă rezultă atunci

$$\varphi_1 = \frac{1}{1 - C_{11}} [f_1 - f_0 + (1 - C_{10}) \varphi_0] = \frac{f_1 + C_{1c} f_0}{1 - C_{11}}$$

Cunoscând pe φ_1 se poate folosi relația de iterare pentru a calcula pe φ_2 , etc.

20. FOLOSIREA SUMELOR ȘI SERIILOR ÎN APROXIMĂRI

20.1. Evaluarea aproximativă a sumelor seriilor numerice convergente

Fie $\sum_{n \geq 0} a_n$ o serie numerică convergentă avînd suma S ,
adică

$$S = a_0 + a_1 + \dots + a_n + a_{n+1} + \dots$$

Să notăm prin S_n suma primilor n termeni și prin R_n suma seriei :

$$R_n = a_{n+1} + a_{n+2} + \dots \quad \text{Evident } S = S_n + R_n.$$

Se obișnuiește adesea să se aproximeze suma exactă S a seriei printr-o suma parțială S_n . Eroarea care se comite în acest fel este măsurată prin numărul R_n numit și restul de ordinul n al seriei. Evident cînd n tinde spre infinit seria fiind convergentă, R_n va tinde spre zero. Aproximarea sumei S prin S_n este utilă dacă se poate aprecia mărimea erorii care se face în acest caz, adică dacă se poate găsi o valoare $\varepsilon > 0$, astfel încît $|R_n| < \varepsilon$.

Se spune în acest caz că S_n reprezintă o ε -aproximare a sumei seriei $\sum_{n \geq 0} a_n$.

Determinarea unei majorante ε pentru restul R_n se face în funcție de tipul seriei numerice. Se iau în acest fel în considerare următoarele cazuri :

Seria este alternantă, adică :

$$S = a_0 - a_1 + a_2 - a_3 + \dots + (-1)^n a_n + \dots ; a_n \geq 0$$

Grupînd termenii în felul următor :

$$S = (a_0 - a_1) + (a_2 - a_3) + \dots + (a_{2n} - a_{2n+1}) + \\ + (a_{2n+2} - a_{2n+3}) + \dots$$

Se observă că $S_{2n+1} < S$, adică orice sumă parțială de ordin impar este mai mică decît suma seriei. Grupînd termenii seriei astfel :

$$S = a_0 - (a_1 - a_2) - (a_3 - a_4) - \dots - (a_{2n-1} - a_{2n}) - \\ - (a_{2n+1} - a_{2n+2}) - \dots$$

se observă că $S_{2n} > S$, adică orice sumă de ordin par este mai mare decît suma seriei. Următoarele inegalități devin în acest caz adevărate :

$$0 < |S_{2n} - S| \leq S_{2n} - S_{2n+1} = a_{2n+1}$$

$$0 < |S - S_{2n+1}| \leq S_{2n+2} - S_{2n+1} = a_{2n+2}$$

Deci în toate cazurile $|R_n| \leq a_{n+1}$, cu alte cuvinte eroarea care se comite atunci cînd se aproximează suma unei serii alternante convergente printr-o sumă parțială de ordinul n , este în valoare absolută mai mică decît primul termen neglijat.

Aplicația 1

Se dă seria armonică alternantă $\ln 2 = \sum_{n \geq 1} (-1)^{n+1} \frac{1}{n}$

Cîți termeni trebuie să fie însumați pentru a obține suma seriei cu trei zecimale exacte, în fapt cu o eroare mai mică decît 0,001 ?

Din cele arătate anterior rezultă că dacă înlocuim suma seriei cu suma parțială de ordinul n , rezultă o eroare în valoare absolută mai mică decît termenul $1/(n+1)$. Deci problema pusă revine la determinarea lui n din inegalitatea $1/(n+1) \leq 0,001$, adică $n > 999$. Deci trebuie să însumăm 999 termeni din seria armonică pentru a obține pe $\ln 2$ cu trei

zecimale exacte. Se spune că această serie este slab convergentă.

Seria $\sum_{n \geq 1} a_n$, convergentă, este cu termeni pozitivi.

Fie $m \in \mathbb{N}$ un rang cu proprietatea că pentru orice $n \geq m$, să avem $a_{n+1}/a_n \leq q < 1$. Rezultă $a_{n+1} \leq qa_n$.

Iterînd această relație, obținem :

$$a_{n+2} \leq qa_{n+1}$$

$$a_{n+3} \leq qa_{n+2} < q^2 a_{n+1}$$

$$\dots \dots \dots$$

Rezultă :

$$R_n = a_{n+1} + a_{n+2} + \dots < a_{n+1}(1 + q + q^2 + \dots)$$

Deci :

$$R_n < \frac{a_{n+1}}{1 - q}$$

Aplicația 2

Se dă seria convergentă $\sum_{n \geq 1} \frac{2^n}{n \cdot 3^n}$

Să se estimeze eroarea care se comite cînd suma seriei este aproximată prin suma primilor 10 termeni.

Se observă că :

$$\frac{u_{n+1}}{u_n} = \frac{2^{n+1}}{(n+1) \cdot 3^{n+1}} \cdot \frac{n \cdot 3^n}{2^n} = \frac{2}{3} \left(\frac{n}{n+1} \right) < \frac{2}{3},$$

pentru orice $n \in \mathbb{N}$.

Rezultă folosind formula de majorare a restului de ordinul 10 :

$$R_{10} < \frac{a_{11}}{1 - 2/3} = 0,004$$

Se consideră în continuare o serie numerică convergentă cu termeni pozitivi $\sum_{n \geq 1} f(n)$ scrisă cu ajutorul unei funcții pozitive

$f: [1, \infty) \rightarrow \mathbb{R}_+$ continuă și descrescătoare. Este evident că în aceste condiții avem :

$$0 \leq f(n+1) \leq \int_n^{n+1} f(x)dx \leq f(n)$$

Rezultă :

$$f(n+1) + f(n+2) + \dots + f(n+p) \leq \int_n^{n+p} f(x)dx$$

Făcînd ca p să tindă spre infinit, se obține :

$$R_n \leq \lim_{p \rightarrow \infty} \int_n^{n+p} f(x)dx.$$

Aplicația 3

Se dă seria convergentă cu termeni pozitivi $\sum_{n \geq 1} \frac{1}{n^2}$

Să se estimeze eroarea care se face cînd se aproximează suma seriei prin suma primilor 10 termeni. Atunci :

$$\sum_{n \geq 1} \frac{1}{n^2} = \sum_{n \geq 1} f(n),$$

unde $f: [1, \infty) \rightarrow \mathbb{R}_+$, definită prin relația $f(x) = 1/x^2$, este pozitivă, continuă și descrescătoare pe intervalul de definiție.

Folosind cele arătate în calculul unei majorante pentru R_n , se obține :

$$R_{10} < \lim_{p \rightarrow \infty} \int_{10}^{10+p} \frac{1}{x^2} dx = \lim_{p \rightarrow \infty} \left(-\frac{1}{p} + \frac{1}{10} \right) = \frac{1}{10} = 0,1$$

20.2. Transformarea seriilor numerice slab convergente în serii mai rapid convergente

O condiție suficientă pentru ca seria numerică $\sum_{n \geq 1} a_n$ să fie mai rapid convergentă decît seria $\sum_{n \geq 1} b_n$, este ca $\lim_{n \rightarrow \infty} (a_n/b_n) = 0$. Una dintre cele mai simple transformări

a unei serii numerice slab convergente într-o serie mai rapid convergentă este transformarea lui Euler. Fie seria convergentă $a_1 - a_2 + a_3 - a_4 + \dots + (-1)^{n-1}a_n + \dots$. Prin transformarea lui Euler seria de mai sus devine :

$$\frac{a_1}{2} - \frac{a_2 - a_1}{2} + \frac{a_3 - a_2}{2} - \frac{a_4 - a_3}{2} + \dots + (-1)^{n+1} \frac{a_n - a_{n-1}}{2} + \dots$$

Seria astfel transformată are aceeași sumă ca și seria inițială. Într-adevăr, dacă S_n și S'_n sînt sumele parțiale de ordin n al celor două serii, avem :

$$S_n - S'_n = \frac{(-1)^n}{2} a_n.$$

Deoarece seria inițială este convergentă, șirul termenilor a_n tinde către zero cînd n tinde spre infinit, deci $S = S'$. Prin transformarea lui Euler unele serii se transformă în serii mai rapid convergente.

Aplicația 4

Să se transforme seria slab convergentă :

$$\ln 2 = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots + (-1)^{n+1} \frac{1}{n} + \dots$$

într-o serie mai rapid convergentă, folosind transformarea lui Euler. Aplicînd o dată transformarea de mai sus obținem :

$$\ln 2 = \frac{3}{4} - \frac{1}{12} + \frac{1}{24} - \dots + \frac{1}{4n(2n-1)} - \frac{1}{4n(2n+1)} + \dots$$

Aplicînd încă o dată transformarea lui Euler avem :

$$\ln 2 = \frac{17}{24} - \frac{1}{48} + \frac{1}{120} - \dots + \frac{1}{4n(4n^2-1)} - \frac{1}{8n(n+1)(2n+1)} + \dots$$

Aplicind pentru a treia oară transformarea lui Euler se obține :

$$\ln 2 = \frac{67}{96} - \frac{1}{160} + \frac{1}{480} - \dots + \frac{3}{8n(4n^2 - 2)(2n + 2)} - \dots$$

Pentru calculul lui $\ln 2$ cu 3 zecimale exacte așa cum s-a văzut în aplicația 1 sînt necesari 999 termeni ai seriei inițiale dar numai 3 termeni ai seriei de trei ori transformată. Într-adevăr, avem :

$$\ln 2 \simeq \frac{67}{96} - \frac{1}{160} + \frac{1}{480} = 0,693.$$

O variantă îmbunătățită a transformării lui Euler, se obține folosind diferențele finite. Astfel suma S a unei serii alternante convergente poate fi scrisă folosind diferențele finite astfel :

$$\begin{aligned} S &= a_0 - a_1 + a_2 - a_3 + \dots (-1)^k a_k + \dots = \\ &= \frac{1}{2} \left[a_0 - \frac{1}{2} \Delta a_0 + \frac{1}{4} \Delta^2 a_0 - \frac{1}{8} \Delta^3 a_0 + \dots + \right. \\ &\quad \left. + \left(-\frac{1}{2} \right)^k \Delta^k a_0 + \dots \right] \end{aligned}$$

Uneori seria cu diferențe finite este mai rapid convergentă decît prima.

Aplicația 5

Să se folosească transformarea lui Euler cu diferențe finite pentru a mări rapiditatea convergenței seriei armonice alternante.

Avem, folosind primii patru termeni ai seriei cu diferențe finite :

$$\begin{aligned} \ln 2 &= 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots + (-1)^{n+1} \frac{1}{n} + \dots \simeq \\ &\simeq \frac{1}{2} \left[1 - \left(\frac{1}{2} - 1 \right) + \frac{1}{4} \left(\frac{1}{3} - 2 \cdot \frac{1}{2} + 1 \right) - \right. \\ &\quad \left. - \frac{1}{8} \left(\frac{1}{4} - 3 \frac{1}{3} + 3 \frac{1}{2} - 1 \right) \right] = 0,681 \end{aligned}$$

Valoarea lui $\ln 2$, exactă pînă la a treia zecimală, este 0,693. Transformarea lui Kummer mărește rapiditatea convergenței seriei $\sum_{n \geq 1} a_n$ folosind o serie $\sum_{n \geq 1} u_n$ avînd o sumă cunoscută U .

Dacă $\lim_{n \rightarrow \infty} (a_n/u_n) = \lambda$ atunci prin transformarea lui Kummer seria

$$\lambda U + \sum_{n \geq 1} (a_n - \lambda u_n)$$

are evident aceeași sumă ca seria inițială $\sum_{n \geq 1} a_n$, în plus este mai rapid convergentă. Într-adevăr, avem :

$$\lim_{n \rightarrow \infty} \frac{a_n - \lambda u_n}{a_n} = \lim_{n \rightarrow \infty} \left(1 - \lambda \frac{u_n}{a_n} \right) = 0.$$

Rezultă că este îndeplinită condiția suficientă ca seria transformată să fie mai rapid convergentă decît seria inițială.

Aplicația 6

Să se mărească rapiditatea convergenței seriei $\sum_{n \geq 1} \frac{1}{n^2}$ folosind criteriul lui Kummer.

Se folosește seria $\sum_{n \geq 1} \frac{4}{4n^2 - 1}$ a cărei sumă poate fi calculată.

Într-adevăr :

$$\begin{aligned} \sum_{n \geq 1} \frac{4}{4n^2 - 1} &= \sum_{n \geq 1} \left(\frac{2}{2n - 1} - \frac{2}{2n + 1} \right) = \left(\frac{2}{1} + \frac{2}{3} + \frac{2}{5} + \right. \\ &\quad \left. + \frac{2}{7} + \dots \right) - \left(\frac{2}{3} + \frac{2}{5} + \frac{2}{7} + \dots \right) = 2 \end{aligned}$$

Avem, de asemenea :

$$\lambda = \lim_{n \rightarrow \infty} \frac{1}{n^2} \cdot \frac{4n^2 - 1}{4} = 1.$$

Deci prin transformarea lui Kummer seria inițială se transformă în seria

$$2 - \sum_{n \geq 1} \frac{1}{n^2(4n^2 - 1)}$$

avînd aceeași sumă ca seria dată, dar mai rapid convergentă.

20.3. Folosirea formulei Taylor și a seriei Taylor în aproximări

Fiind dată funcția $f: (\alpha, \beta) \rightarrow \mathbb{R}$ de $n + 1$ ori derivabilă pe (α, β) , iar $a \in (\alpha, \beta)$, atunci pentru orice $x \in (\alpha, \beta)$, există un punct ξ cuprins între a și x , astfel încît să fie adevărată formula lui Taylor

$$f(x) = f(a) + \frac{(x-a)}{1!} f'(a) + \dots + \frac{(x-a)^n}{n!} f^n(a) + \\ + \frac{(x-a)^{n+1}}{(n+1)!} f^{(n+1)}(\xi); \quad \xi \in (a, x).$$

În formula anterioară ultimul termen notat $R_n(x)$ reprezintă restul lui Lagrange :

$$R_n(x) = \frac{(x-a)^{n+1}}{(n+1)!} f^{(n+1)}(\xi), \quad \xi \in (a, x)$$

De obicei se rescrie restul R_n sub forma :

$$R_n(x) = \frac{(x-a)^{n+1}}{(n+1)!} f^{(n+1)}[a + \theta(x-a)], \quad 0 < \theta < 1.$$

În aceleași ipoteze asupra funcției f , dacă $0 \in (\alpha, \beta)$, atunci există $0 < \theta < 1$, astfel încît :

$$f(x) = f(0) + \frac{x}{1!} f'(0) + \dots + \frac{x^n}{n!} f^{(n)}(0) + \frac{x^{n+1}}{(n+1)!} f^{(n+1)}(\theta x)$$

Formula de mai sus poartă numele lui Mac-Laurin. Deoarece primii n termeni ai formulei lui Taylor formează un polinom

de gradul n , notat $T_n(x)$, formula lui Taylor poate fi scrisă :

$$f(x) = T_n(x) + R_n(x).$$

Pentru $n \geq 1$, avem :

$$f(x) - T_n(x) = f(x) - T_{n-1}(x) - \frac{(x-a)^n}{n!} f^{(n)}(a)$$

Dar din formula lui Taylor de ordinul $n-1$, pentru orice $x \in (\alpha, \beta)$, există $\xi \in (a, x)$, astfel încît :

$$f(x) - T_{n-1}(x) = \frac{(x-a)^n}{n!} f^{(n)}(\xi)$$

Rezultă :

$$f(x) - T_n(x) = \frac{(x-a)^n}{n!} [f^{(n)}(\xi) - f^{(n)}(a)]$$

Deci :

$$\frac{f(x) - T_n(x)}{(x-a)^n} = \frac{1}{n!} [f^{(n)}(\xi) - f^{(n)}(a)]$$

Atunci cînd $x \rightarrow a$, rezultă $\xi \rightarrow a$ și deci $\lim_{n \rightarrow \infty} \frac{f(x) - T_n(x)}{(x-a)^n} = 0$.

Formula lui Taylor împreună cu relația de mai sus pot fi utilizate în calculul aproximativ al unor mărimi, în calculul limitelor, etc.

Aplicația 7

Să se folosească formula lui Taylor cu restul de ordinul unu pentru a calcula aproximativ $\sqrt[4]{260}$.

Se consideră funcția $f(x) = \sqrt[4]{x}$. Formula lui Taylor cu restul de ordinul unu pentru această funcție este :

$$\begin{aligned} f(x) &= f(a) + \frac{(x-a)}{1!} f'(a) + \frac{(x-a)^2}{2!} f''(\xi) = \\ &= \sqrt[4]{a} + \frac{(x-a)}{1!} \frac{1}{4 \sqrt[4]{a^3}} - \frac{(x-a)^2}{2!} \frac{3}{4^2} \frac{1}{\sqrt[4]{\xi^7}}, \quad \xi \in (a, x) \end{aligned}$$

Luăm $x = 260$, $a = 256$. Se obține folosind primii doi termeni din formula de mai sus :

$$\sqrt[4]{260} \simeq 4 + \frac{4}{1!} \frac{1}{4^4} = 1,0156.$$

Eroarea comisă este dată de

$$|R_1(\xi)| = \left| \frac{4^2}{2!} \frac{3}{4^2} \frac{1}{\sqrt[4]{\xi^7}} \right|, \quad \xi \in (256, 260)$$

Alegând $\xi = 256$, se obține $|R_1| < 0,00009$.

Aplicația 8

Folosind formula Mac-Laurin cu restul de ordinul 2, să se calculeze

$$I = \lim_{x \rightarrow \infty} \left[x - x^2 \ln \left(1 + \frac{1}{x} \right) \right]$$

Notînd $x = 1/y$, avem :

$$I = \lim_{x \rightarrow \infty} \left[x - x^2 \ln \left(1 + \frac{1}{x} \right) \right] = \lim_{y \rightarrow 0} \left[\frac{1}{y} - \frac{1}{y^2} \ln(1 + y) \right]$$

Folosind formula lui Mac-Laurin cu restul de ordinul 2 pentru funcția $\ln(1 + y)$, se obține :

$$\begin{aligned} I &= \lim_{y \rightarrow 0} \left[\frac{1}{y} - \frac{1}{y^2} \ln(1 + y) \right] = \lim_{y \rightarrow 0} \left[\frac{y - \ln(1 + y)}{y^2} \right] = \\ &= \lim_{y \rightarrow 0} \frac{y - \left[y - \frac{y^2}{2} + R_2(0y) \right]}{y^2} = \frac{1}{2}, \text{ deoarece } \lim_{y \rightarrow 0} \frac{R_2(0y)}{y^2} = 0 \end{aligned}$$

Aplicația 9

Să se rezolve aproximativ ecuația $\cos x - 2x^2 = 0$ folosind formula lui Mac-Laurin cu restul de ordinul doi.

Se înlocuiește în ecuație funcția $\cos x$ prin primii doi termeni ai dezvoltării sale după formula lui Mac-Laurin neglijînd restul de ordinul doi :

$$1 - \frac{1}{2} x^2 - 2x^2 = 0 \text{ sau } 2 - 5x^2 = 0$$

Rezultă : $x_{1,2} \simeq \pm \sqrt{10/5} = \pm 0,632$

Calculul numeric al rădăcinilor conduce, folosind metoda lui Newton, la valorile $\pm 0,635$.

Se consideră în continuare o funcție f definită pe un interval (α, β) și indefinit derivabilă în punctul $a \in (\alpha, \beta)$. Dacă șirul $R_n(x)$ al resturilor din formula lui Taylor scrisă pentru funcția f este convergent către zero pe un interval $I \subset (\alpha, \beta)$, adică :

$$\lim_{n \rightarrow \infty} R_n(x) = 0, x \in I \subset (\alpha, \beta)$$

atunci seria de puteri ale lui $(x - a)$

$$f(a) + \frac{(x-a)}{1!} f'(a) + \frac{(x-a)^2}{2!} f''(a) + \dots + \frac{(x-a)^n}{n!} f^{(n)}(a) + \dots$$

numită *seria Taylor a funcției f în punctul a* este convergentă pentru $x \in I \subset (\alpha, \beta)$ către $f(x)$, deci

$$f(x) = f(a) + \frac{(x-a)}{1!} f'(a) + \dots + \frac{(x-a)^n}{n!} f^{(n)}(a) + \dots, x \in I$$

Formula de mai sus reprezintă dezvoltarea funcției f în serie Taylor în jurul punctului a .

Pentru $a = 0$ se obține dezvoltarea funcției f în serie Mac-Laurin în jurul punctului 0.

$$f(x) = f(0) + \frac{x}{1!} f'(0) + \dots + \frac{x^n}{n!} f^{(n)}(0) + \dots, x \in I$$

Se dau în continuare dezvoltările în serie Mac-Laurin ale unor funcții elementare punând în evidență și intervalele de convergență pentru fiecare serie.

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}, x \in \mathbb{R}$$

$$\sin x = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!}, x \in \mathbb{R}$$

$$\cos x = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!}, \quad x \in \mathbb{R}$$

$$\ln(1+x) = \sum_{n=1}^{\infty} (-1)^{n+1} \frac{x^n}{n}, \quad |x| < 1$$

$$(1+x)^\lambda = 1 + \sum_{n=1}^{\infty} \frac{\lambda(\lambda-1)\dots(\lambda-n+1)}{n!} x^n, \quad |x| < 1, \lambda \in \mathbb{R}$$

Aplicația 10

Să se calculeze aproximativ $I = \int_0^1 \frac{\sin x}{x} dx$ folosind dezvoltarea în serie Mac-Laurin a funcției $\sin x$.

Integrala nu poate fi calculată exact deoarece nu se poate pune în evidență o primitivă exprimată prin funcții elementare. Avem :

$$\int_0^1 \frac{\sin x}{x} dx = \int_0^1 \frac{1}{x} \left(x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \right) dx =$$

$$= \int_0^1 \left(1 - \frac{x^2}{3!} + \frac{x^4}{5!} - \frac{x^6}{7!} + \dots \right) dx =$$

$$= \left(x - \frac{x^3}{3 \cdot 3!} + \frac{x^5}{5 \cdot 5!} - \frac{x^7}{7 \cdot 7!} + \dots \right) \Big|_0^1 =$$

$$= 1 - \frac{1}{3 \cdot 3!} + \frac{1}{5 \cdot 5!} - \frac{1}{7 \cdot 7!} + \dots$$

Folosind primii trei termeni se obține $I \simeq 0,94611$.

Eroarea care se face prin aproximare, seria fiind alternantă, este mai mică în valoare absolută decât primul termen neglijat al seriei, deci mai mică decât 0,000028.

21. METODE PROBABILISTICE

21.1. Introducere

Metode probabilistice (sau metode Monte Carlo) sînt acele procedee destinate unor clase largi de probleme deterministe (de exemplu rezolvarea integralelor, determinarea valorilor extreme, rezolvarea ecuațiilor ordinare sau cu derivate parțiale) care fac uz de elemente aleatoare. La baza metodelor probabilistice stă posibilitatea de a genera numere aleatoare (denumite de multe ori pseudoaleatoare) prin formule de recurență deterministe. De exemplu, folosind formula $x_{k+1} = s \cdot x_k \pmod{m}$, cu $s = 13$, $m = 100$, $x_0 = 1$, obținem următoarea secvență de numere formate din două cifre: 01, 13, 69, 97, 61, 93, 09, 17, 21, 73, 49, 37, 81, 53, 89, 57, 41, 33, 29, 77. După numărul 77, secvența de numere continuă din nou cu 1.

Deși nu este nimic aleator în modul în care au fost obținute aceste numere, ele au toate caracteristicile numerelor aleatoare. Dacă le reprezentăm pe o axă în intervalul $[1, 100]$, se observă o distribuție uniformă a acestora fără nici o preferință față de o parte a intervalului. Dacă se împart toate numerele de mai sus la 100, se obține o nouă secvență de numere pseudoaleatoare care vor lua valori în intervalul $[0,1]$. O secvență de numere pseudoaleatoare mult mai cuprinzătoare poate fi obținută cu formula de recurență: $x_{k+1} = 7^9 \cdot x_k \pmod{10^r}$, $x_0 = 1$, unde în mod

obișnuit se alege $r = 10$. Modul în care o astfel de secvență de numere pseudoaleatoare poate fi utilizată pentru rezolvarea unor probleme deterministe este cel mai bine ilustrat de calculul aproximativ al integralei $\int_0^1 f(x)dx$, presupunând f continuă pe $[0,1]$.

21.2. Procedee bazate pe valoarea medie

Conform acestui procedeu se evaluează funcția integrant într-un număr de puncte pseudoaleatoare uniform repartizate în intervalul $[0,1]$. Atunci o valoare aproximativă pentru integrala $\int_0^1 f(x)dx$ se obține înmulțind valoarea medie $M[f(x)]$ cu lungimea intervalului de integrare.

Deoarece o estimare a valorii medii o reprezintă media aritmetică, rezultă următoarea formulă de calcul aproximativ :

$$\int_0^1 f(x)dx \simeq \frac{\sum_{i=1}^n f(x_i)}{n}$$

Pentru cazul particular $f(x) = x$ și folosind primele șapte numere (împărțite în prealabil cu 100) din secvența de numere pseudoaleatoare dată, se obține :

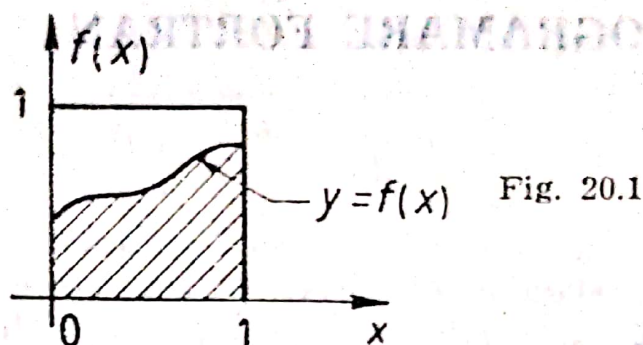
$$\int_0^1 x dx \simeq \frac{0,01 + 0,13 + 0,69 + 0,97 + 0,61 + 0,93 + 0,09}{7} = 0,49$$

valoarea exactă fiind 0,50.

21.3. Procedee bazate pe simulare

Să presupunem că funcția integrant este mărginită și ea a fost astfel transformată încât să verifice condiția $0 \leq f(x) \leq 1$.

Reprezentind grafic in acest caz funcția $f(x)$ in interiorul pătratului de latură egal cu 1, este evident că valoarea integralei $\int_0^1 f(x)dx$ este egală cu mărimea ariei hașurate din figura 20.1, conform interpretării geometrice a integralei definite.



Dacă se presupune suprafața pătratului acoperită uniform cu n puncte (ale căror coordonate sînt perechi de numere aleatoare din secvența dată împărțite în prealabil cu 100) și dacă m dintre aceste puncte cad în porțiunea hașurată, este evident că raportul $\frac{m}{n}$ va reprezenta o aproximare a integralei date. Se simulează de fapt activitatea de tragere la țintă (fiecare punct din interiorul pătratului avînd o probabilitate egală de a fi nimerit) și numărînd apoi numărul m al loviturilor din partea hașurată. Pentru ca raportul m/n să reprezinte o bună aproximare pentru valoarea integralei, trebuie ca n să fie suficient de mare, de obicei de ordinul 10^4 .

22. ELEMENTE ALE LIMBAJULUI DE PROGRAMARE FORTRAN

22.1. Introducere

Limbajul de programare FORTRAN, folosit în prezent de toate calculatoarele mijlocii sau mari a apărut în anul 1956 și reprezintă o abreviere de la FORmula TRANslator. Este destinat în special pentru rezolvarea problemelor științifice sau ingineresti.

Prima versiune de succes a limbajului este FORTRAN II. Acesta a fost ulterior extins și perfecționat fiind intitulat FORTRAN IV. Pentru calculatoarele foarte mari se pot utiliza și versiuni mai eficiente ale limbajului FORTRAN, și anume FORTRAN V.

În general, orice versiune inferioară este compatibilă cu o versiune superioară. De exemplu, un program scris în FORTRAN II este acceptat și înțeles de către un calculator ce lucrează cu versiunea FORTRAN IV, dar nu și invers. O restricție a limbajului FORTRAN IV o constituie FORTRAN IV BASIC, foarte des utilizat.

În cele ce urmează se prezintă un minimum de cunoștințe, privind limbajul FORTRAN IV, necesare înțelegerii subrutinelor date în această lucrare.

Un program în FORTRAN trebuie scris pe formulare de programare ce cuprind 20 linii și 80 coloane. Fiecare linie din formularul de programare este destinată reprezentării unei informații pe primele 72 de poziții, ultimele 8 po-

ziții fiind rezervate comentariilor. Dintre cele 72 de poziții active primele 5 sînt rezervate etichetelor instrucțiunilor care trebuie scrise aliniate spre dreapta.

Poziția a 6-a este rezervată (scriind de obicei o cifră diferită de zero) pentru a se indica dacă este nevoie, că o instrucțiune se continuă și pe linia următoare. Rezultă că toate instrucțiunile FORTRAN trebuie scrise începînd cu poziția a 7-a. De exemplu, instrucțiunea următoare care atribuie variabilei A valoarea $B + C$

5	6	7
12		$A = B + C$

este scrisă în formularul de programare începînd cu poziția a 7-a și are eticheta 12 scrisă în primele 5 poziții, fiind aliniată spre dreapta în cadrul acestei zone. Lipsa oricărui caracter în poziția a 6-a semnalează faptul că instrucțiunea scrisă, fiind scurtă, nu este necesar să fie continuată pe linia următoare.

Dacă se dorește să se folosească un rînd al formularului de programare pentru comentarii, avînd semnificații doar pentru programator, se poate realiza acest lucru scriînd mai întîi în poziția 1 a liniei litera C.

Alfabetul FORTRAN folosește 26 caractere literale: A ÷ Z (alfabetul englez), 4 simboluri aritmetice și anume +, -, *, / și 8 simboluri de punctuație ., ', =, (,), \$, spațiile între caractere.

Constantele întregi se scriu numai cu ajutorul cifrelor. Dacă în cazul numerelor pozitive prezența semnelui + în fața sa este facultativă, în cazul numerelor negative prezența semnelui - în fața este obligatorie. Constantele reale se scriu obligatoriu cu ajutorul punctului zecimal care are același rol ca virgula folosită în aritmetică. De exemplu, 0.31 reprezintă de fapt valoarea 0,31. În cea mai simplă formă de reprezentare a constantelor reale se pot folosi maximum 7 cifre semnificative. Variabilele FORTRAN sînt mărimi ale căror valori se modifică în cursul executării programelor. Pentru reprezentarea variabilelor se folosesc identificatori. Un identificator este un grup de 1-6 caractere (litere sau cifre) dintre care primul este în mod obli-

gatoriu o literă. Exemple de variabile scrise corect *X*, *Y2*, *NPLUSM* și greșit *2M* (începe cu o cifră în loc de o literă) *MATRICE* (conține 7 litere, număr nepermis de mare). Există o convenție în limbajul FORTRAN conform căreia identificatorii care încep cu una din literele *I*, *J*, *K*, *L*, *M*, *N* sînt de tip întreg, în caz contrar sînt considerați de tip real. Exemple de variabile de tip întreg *N*, *M2*, *ION*. Exemple de variabile de tip real *X*, *Y4*, *ZET*, *EPS*.

Pentru a putea interpreta variabilele algebrice cu indici, limbajul FORTRAN folosește variabilele indexate. Să presupunem de exemplu se cere calculul sumei

$$S = \sum_{k=1}^{100} x_k = x_1 + x_2 + \dots + x_{100}.$$

Pentru a nu folosi 100 variabile distincte, fie acestea *X1*, *X2*, ..., *X100*, limbajul FORTRAN permite să se folosească un nume comun pentru toate aceste variabile, ele deosebindu-se prin indici diferiți. Deci se poate defini variabila indexată *X* cu 100 componente și anume *X(1)*, *X(2)*, ..., *X(100)*. Utilizarea în program a acestei variabile indexate *X* nu este permisă decît dacă la începutul programului (decî înainte de folosirea ei); ea apare într-o declarație de dimensiune de tipul :

```

5 | 6 | 7
-----
| | DIMENSION X (100)

```

Valoarea minimă pe care poate să o ia un indice în limbajul FORTRAN este 1 (decî nu zero cum se obișnuiește în algebră). Pot fi utilizate și variabile indexate avînd doi sau mai mulți indici. De exemplu *Y(3,4)* este o componentă a variabilei indexate numită *Y* și anume componenta care în tabelul de variabile (corespunzător unei matrici din algebră) ocupă poziția situată la intersecția liniei a 3-a cu coloana a 4-a. Presupunînd că variabila *Y* are 10 linii și 10 coloane (corespunzînd decî unei matrici pătrate), ea trebuie declarată ca dimensiune la începutul programului astfel :

```

5 | 6 | 7
-----
| | DIMENSION Y(10, 10)

```


Indicii unei variabile indexate pot fi nu numai numere întregi dar și variabile de tip întreg sau expresii formate cu acestea. De exemplu $P(J, K)$; $V1(1, 2 * K + 1)$.

22.2. Principalele instrucțiuni ale limbajului FORTRAN

a) Instrucțiuni de atribuire aritmetică

Aceste instrucțiuni sînt utilizate în FORTRAN în vederea programării algoritmilor ce conțin calcule de tip algebric. Să presupunem de exemplu că într-un algoritm dat intervine la un moment dat secvența $x = \alpha + \beta$. Această secvență este preluată și programată în FORTRAN cu ajutorul unei instrucțiuni de atribuire astfel :

5	6	7	
			$X = ALFA + BETA$

Se observă deci că s-au notat variabilele x, α, β prin $X, ALFA$, respectiv $BETA$, deoarece limbajul FORTRAN nu folosește literele mici ale alfabetului sau ale alfabetului grecesc. Sensul acestei instrucțiuni de tip aritmetic este următorul :

Se evaluează expresia aritmetică din partea dreaptă a semnului egal, după care valoarea găsită este „atribuită” variabilei din partea stîngă a semnului egal, deci lui X . Această operație de „atribuire” explică de ce pot fi scrise instrucțiuni de tipul :

5	6	7	
			$X = X + 1$

care în algebră nu au sens.

Într-o instrucțiune de atribuire aritmetică în partea stîngă a semnului egal se găsește întotdeauna o variabilă, iar în partea dreaptă o expresie aritmetică.

Expresiile aritmetice se formează cu operatorii aritmetici

Simbol	Exemplu
+	$A + B$
-	$A - B$
*	$A * B$ (A înmulțit cu B)
/	A / B (A împărțit la B)
**	$A ** 2$ (A la puterea a doua)

Pentru a codifica corect expresiile aritmetice trebuie să se țină cont de regulile de prioritate ale operațiilor. Într-o expresie aritmetică care nu conține paranteze, operațiile sînt executate în ordinea următoare (priorități naturale):

- 1) calcularea valorii funcției
- 2) ridicări la putere
- 3) înmulțiri și împărțiri (prioritate egală)
- 4) adunări și scăderi (prioritate egală)

Prin introducerea parantezelor (care în FORTRAN sînt numai rotunde) se poate impune o altă ordine de executare a operațiilor. Într-o expresie cu paranteze se vor executa mai întîi operațiile cuprinse între parantezele situate interior, iar în interiorul acestora se vor respecta prioritățile naturale. În scopul facilitării activității de programare pentru o serie de funcții elementare frecvent întîlnite în calcule, ca de exemplu e^x , $\sin x$, $\cos x$, $\ln x$, $\arctg x$, $\sqrt{x}$ etc. se folosește în limbajul FORTRAN așa-numitele *funcții aritmetice standard* (sau *funcții de bibliotecă*).

Pentru funcțiile cel mai des întîlnite, codificările sînt:

e^x	—	EXP (X)
$\ln x$	—	ALOG (X)
$\log_{10} x$	—	ALOG 10 (X)
$\sin x$	—	SIN (X)
$\cos x$	—	COS (X)
$\arctg x$	—	ATAN (X)
$\sqrt{x}$	—	SQRT (X)

De exemplu, dacă în decursul unui program se dorește să se calculeze rădăcina pătrată din variabila reală X , iar

rezultatul operațiunii să fie atribuit variabilei reale Y , este suficient să scriem instrucțiunea de atribuire :

5	6	7
		$Y = \text{SQRT}(X)$

Două variabile se pot compara folosind operatori de relație. Aceștia sînt :

Operator	Exemple	Scriere algebrică
	Scriere FORTRAN	
.EQ.	$A.EQ.B$	$A = B$
.NE.	$A.NE.B$	$A \neq B$
.LT.	$A.LT.B$	$A < B$
.LE.	$A.LE.B$	$A \leq B$
.GT.	$A.GT.B$	$A > B$
.GE.	$A.GE.B$	$A \geq B$

b) Instrucțiunea de salt necondiționat

În general programul FORTRAN are o desfășurare secvențială, instrucțiune cu instrucțiune, pe măsura apariției acestora.

Introdusă în programul de calcul atunci cînd îi vine rîndul să fie executată, instrucțiunea de salt necondiționat are ca efect transferul controlului desfășurării programului la o altă instrucțiune din program (deci, nu la următoarea) avînd o etichetă identică cu aceea care urmează după cuvîntul cheie **GO TO**.

Exemplu :

5	6	7
		$I = I + 1$
		GO TO 14
25		$A = B - C$
14		$Y = \text{SIN}(X)$

Atunci cînd îi vine rîndul să fie executată, instrucțiunea de salt necondiționat produce ignorarea instrucțiunii cu

eticheta 25, în schimb se execută instrucțiunea cu eticheta 14. Este obligatoriu ca instrucțiunea care urmează după **GO TO** să aibă o etichetă, altfel nu există nici o posibilitate de a mai fi executată vreodată.

c) Instrucțiunea de salt condiționat **IF** aritmetic

Forma generală a acestei instrucțiuni este :

5	6	7
		IF (expresie aritmetică) eticheta 1, eticheta 2, eticheta 3

Exemplu :

5	6	7
		:
15		$A = B + C$
		IF $(X - Y) = 15, 20, 25$
20		$K = K + 1$
		$EPS = \sin(A)$

În momentul executării acestei instrucțiuni se evaluează mai întâi expresia aritmetică conținută în paranteze, adică $X - Y$. În funcție de valoarea găsită se efectuează apoi instrucțiunea cu eticheta 15, dacă valoarea este negativă, sau instrucțiunea cu eticheta 20, dacă expresia $X - Y$ este zero, sau instrucțiunea cu eticheta 25, dacă valoarea este pozitivă.

d) Instrucțiunea de salt condiționat **IF** logic

Forma generală este :

5	6	7
		IF (expresia logică) instrucțiune executabilă

Exemplu :

5	6	7
		IF $(I. LT. J)$ GO TO 5

Execuția se face astfel : se determină valoarea de adevăr a expresiei logice din paranteze. Dacă I este mai mic decât J , deci valoarea expresiei este „adevărată“, se trece la execuția instrucțiunii scrise după cuvântul cheie, în cazul de față, **GO TO** 5, apoi se continuă parcurgerea listei de instrucțiuni. Dacă I nu este mai mic decât J , atunci valoarea de

adevăr a expresiei logice este „falsă”, se ignoră instrucțiunea **GO TO 5**, trecându-se direct la executarea următoarei instrucțiuni din program.

e) Instrucțiunile **STOP** și **END**

Sînt de obicei ultimele două instrucțiuni dintr-un program independent, în ordinea aceasta, semnalînd terminarea programului de calcul.

f) Instrucțiunile **DO** și **CONTINUE**

Aceste instrucțiuni în ordinea dată semnalează începutul și sfîrșitul unui ciclu de calcule.

În mod obișnuit formăm un ciclu de calcul utilizînd instrucțiunile **GO** și **CONTINUE** astfel:

5 6 7	DO n $K = n_1, n_2, n_3$ CONTINUE
-----------	--

unde: n reprezintă eticheta ultimei instrucțiuni din ciclu în cazul de față instrucțiunea **CONTINUE**.

K reprezintă indicele contor al buclei

n_1 reprezintă valoarea inițială a indicelui

n_2 reprezintă valoarea finală a indicelui

n_3 reprezintă rata de schimbare a indicelui (pas de variație)

Dacă valoarea n_3 lipsește, se ia automat pasul de variație egal cu 1.

Exemplu :

5 6 7	DO 5 $K = 1, 20$ $X(K) = 0.$ $Y(K) = 0.$ 5 CONTINUE
-----------	--

Execuția acestui ciclu de calcul se face în felul următor: se atribuie mai întîi variabilei contor al ciclului valoarea inițială 1, apoi pentru această valoare se execută toate instrucțiunile din interiorul ciclului, în cazul de față, două.

Rezultă $X(1) = 0$, $Y(1) = 0$. Se transferă din nou controlul la începutul ciclului, mărindu-se valoarea variabilei contor cu o unitate. Se obține $K = 2$. Se compară această valoare cu valoarea finală 20. Deoarece valoarea finală evident nu a fost atinsă, se execută cele două instrucțiuni din interiorul ciclului, obținându-se $X(2) = 0$, $Y(2) = 0$. Se trece din nou controlul la începutul ciclului, mărindu-se valoarea lui cu o unitate, deci $K = 3$. Operațiile sînt reluate pînă cînd valoarea lui K mărită depășește valoarea finală 20. În acest moment controlul execuției programului trece la prima instrucțiune care succede linia finală a ciclului.

Instrucțiunea **CONTINUE** nu este obligatoriu a fi folosită pentru a marca sfîrșitul unui ciclu. Ea poate fi în general înlocuită de orice instrucțiune executabilă, cu excepția următoarelor instrucțiuni :

DO, IF aritmetic, IF logic, GO TO, STOP, PAUSE, RETURN, CALL.

Se acceptă în FORTRAN și două sau mai multe cicluri suprapuse avînd grijă ca incicii contor pentru aceste cicluri să fie notați diferit, dar folosind variabile de tip întreg. În cazul ciclurilor suprapuse, execuția instrucțiunilor este următoarea : se începe cu execuția ciclului exterior ; în momentul cînd se întîlnește un ciclu interior, aceasta se execută complet și apoi se continuă execuția ciclului exterior ; dacă ieșirea forțată din ciclu este posibilă (folosind o instrucțiune de salt condiționat sau necondiționat), intrarea forțată nu este permisă.

g) Instrucțiunile **WRITE** și **FORMAT**

Permit scrierea la imprimantă a rezultatelor execuției programului de calcul. Forma generală a unei instrucțiuni de scriere cu format este următoarea :

5	6	7
m	WRITE (n , m) lista de variabile FORMAT (listă de coduri)	

unde : n este numărul simbolic al imprimantei ; în multe centre de calcul are valoarea 108

m reprezintă eticheta instrucțiunii **FORMAT** atașată instrucțiunii de scriere **WRITE**

variabilei ce urmează să fie scrise, iar n numărul de poziții rezervat zecimalelor.

Limbajul FORTRAN acceptă și alte coduri decât cele amintite, în scopul diversificării posibilităților de imprimare a rezultatelor.

h) Instrucțiunile **READ** și **FORMAT**

Forma generală a unei instrucțiuni de citire cu format este următoarea :

5	6	7
m		READ (n, m) listă de variabile) FORMAT (listă de coduri)

unde : n este numărul simbolic al cititorului de cartele;
În multe centre de calcul are valoarea 105.

m reprezintă eticheta instrucțiunii **FORMAT** atașată instrucțiunii de scriere **READ**

Exemplu :

5	6	7
1		READ (105, 1) ($X(K), K = 1, 10$) FORMAT (10F8.5)

Instrucțiunea de citire **READ** determină citirea de la cititorul de cartele a variabilei indexate X .

Instrucțiunea **FORMAT** atașată lui **READ** indică modul cum trebuie făcută citirea și anume : se citesc cele 10 variabile $X(1), X(2), \dots, X(10)$, fiecare de pe un câmp de 10 poziții ale cartelei, ultimele 5 poziții fiind rezervate zecimalelor.

Aplicația 1

Se prezintă un program de analiză armonică în vederea calculelor efectuate asupra curbelor obținute pe cale experimentală.

În fig. 22.1 se prezintă o curbă de bielă ce poate fi realizată de un mecanism patrulater, fiind reprodusă dintr-un catalog de curbe de bielă (fig. 22.1).

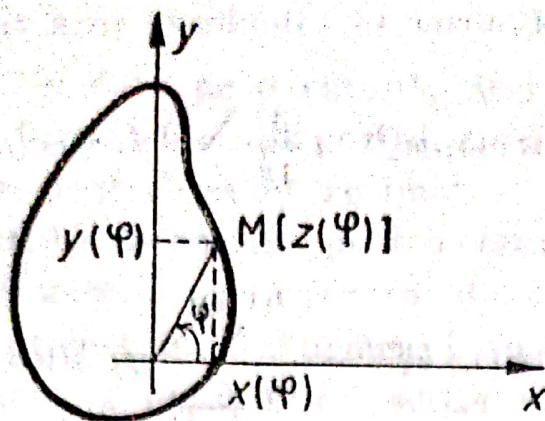


Fig. 22.1

Considerînd un sistem de coordonate în planul complex, un punct oarecare M de pe curbă este de a fix :

$$z(\varphi) = x(\varphi) + iy(\varphi)$$

Măsurînd proiecțiile $x(\varphi)$ și $y(\varphi)$ în 36 puncte egal distanțate ale intervalului $[0, 2\pi]$ în care ia valori unghiul vectorului de poziție φ , se poate forma tabela datelor inițiale ale programului de calcul :

φ_i	$0 = \varphi_1, \varphi_2, \dots, \varphi_{36} = 2\pi$			
$x(\varphi_i)$	x_1	x_2	$\dots$	x_{36}
$y(\varphi_i)$	y_1	y_2	$\dots$	y_{36}

Folosind tabela de valori de mai sus, funcțiile reale $x(\varphi)$ și $y(\varphi)$ pot fi aproximate prin sume parțiale de serii Fourier de forma :

$$x(\varphi) = \frac{a_0^{(1)}}{2} + \sum_{k=1}^{17} (a_k^{(1)} \cos k\varphi + b_k^{(1)} \sin k\varphi) + \frac{1}{2} a_{18}^{(1)} \cos 18\varphi$$

$$y(\varphi) = \frac{a_0^{(2)}}{2} + \sum_{k=1}^{17} (a_k^{(2)} \cos k\varphi + b_k^{(2)} \sin k\varphi) + \frac{1}{2} a_{18}^{(2)} \cos 18\varphi$$

În acest caz coeficienții Fourier se calculează prin relațiile :

$$a_n^{(1)} = \frac{1}{18} \sum_{k=1}^{36} x_k \cos(n\varphi_k), \quad a_n^{(2)} = \frac{1}{18} \sum_{k=1}^{36} y_k \cos(n\varphi_k)$$

$$n = 0, 1, \dots, 18 \quad n = 0, 1, \dots, 18$$

$$b_n^{(1)} = \frac{1}{18} \sum_{k=1}^{36} x_k \sin(n\varphi_k), \quad b_n^{(2)} = \frac{1}{18} \sum_{k=1}^{36} y_k \sin(n\varphi_k)$$

$$n = 1, 2, \dots, 17 \quad n = 1, 2, \dots, 17$$

Se urmărește să se aproximeze $z(\varphi)$ printr-o sumă parțială a unei serii Fourier complexe de forma :

$$z(\varphi) = \sum_{k=-18}^{18} c_k e^{ik\varphi}$$

Aceasta deoarece pe de o parte seriile Fourier complexe sînt mai potrivite spre a fi folosite pentru a aproxima mărimi complexe, iar pe de altă parte aceste serii admit forme simetrice ce pot fi manevrate cu mai multă ușurință în calcule.

Ținînd seama de relațiile lui Euler, se pot pune în evidență formule ce leagă coeficienții c_k de coeficienții a_k , respectiv b_k :

$$c_k = \frac{(a_k^{(1)} + b_k^{(2)}) + (a_k^{(2)} + b_k^{(1)})i}{2}$$

$$c_{-k} = \frac{(a_k^{(1)} - b_k^{(2)}) + (a_k^{(2)} - b_k^{(1)})i}{2}$$

$$c_{18} = c_{-18} = \frac{a_{18}^{(1)} + a_{18}^{(2)}i}{4}$$

$$k = 1, 2, \dots, 17$$

Datele de intrare se normalizează prin împărțire la valoarea cea mai mare și reținînd în continuare cinci zecimale ; se introduce în calculator folosind opt cartele, primele patru pentru proiecțiile x , următoarele patru pentru proiecțiile y .

Se scriu 10 date pe o cartelă, deci fiecareia i se afectează un câmp format din opt poziții, cinci poziții pentru zecimale, o poziție pentru punct zecimal, o poziție pentru valoarea întregului și o poziție pentru semn. Primele trei cartele conțin 10 date, a patra conține doar 6 date (pentru a epuiza valorile proiecțiilor x), urmează apoi 3 cartele cu zece date și în sfârșit a opta avînd scrise numai șase date (pentru proiecțiile y).

Se dă în continuare lista programului de calcul

C PROGRAM PENTRU ANALIZA ARMONICĂ EXPERIMENTALĂ

C

C

INTEGER R

REAL ICØ, ICP, ICN

DIMENSION S1(18), S2(18), U1(18), U2(18),

1 W1(18), W2(18), RCP(18), ICP(18), RCN(18),

2 V(36), X(36), Y(36), PAC(36), TAC(36), ICN(36)

C

C

FORMATE PENTRU INTRARE-IEȘIRE

C

1 FORMAT (10F8.5)

2 FORMAT (///' ANALIZA ARMONICĂ

1 EXPERIMENTALĂ')

3 FORMAT ('xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

2 xxx')

4 FORMAT (///' REPRODUCEREA DATELOR DE

3 INTRARE'/)

5 FORMAT (' X(', I3, ')=' , F9.5, 10X, ' Y(', I3, ')=' ,

4 F9.5)

6 FORMAT (///' COEFICIENTII FOURIER COMPLEXI

5 SINT'/)

7 FORMAT ('COEFICIENTUL (0)=' , F9.5, '+', 1X,

6 F9.5, 'J'/)

8 FORMAT ('COEFICIENTUL (' , I3, ')=' , F9.5, '+',

7 F9.5, 'J', 8X, 'COEFICIENTUL (' , I3, ')=' , F9.5, '+',

8 F9.5, 'J')


```

C
C      CITIREA DATELOR DE INTRARE
C
      READ (1, 1) (X(I), I = 1,36)
C
C      INIȚIALIZĂRI
C
      DO 9 I = 1,36
        PAC(I) = I
        TAC(I) = -I
      9   V(I) = X(I)
      DO 10 I = 1,18
        S1(I) = 0
      10  S2(I) = 0
        S0 = 0
        R = 1
C
C      CALCULUL COEFICIENȚILOR FOURIER
C
      500 DO 11 M = 1,17
        DO 11 I = 1,36
          S1(M) = S1(M) + 1./36.* V(I)* COS(M* I* 3.14159/18.)
      11  S2(M) = S2(M) + 1./36.* V(I)* SIN(M* I* 3.14159/18.)
        DO 12 I = 1,36
      12  S1(18) = S1(18) + 1./72.* V(I)* COS(I* 3.14159)
          S2(18) = 0
          IF (R-2) 100 200, 100
      100 DO 13 I = 1,18
          U1(I) = S1(I)
      13  (U2(I) = S2(I)
          U0 = S0
          DO 14 I = 1,36
      14  V(I) = Y(I)
          DO 14 I = 1,18
            S1(I) = 0
      15  S2(I) = 0
            S0 = 0
            R = R + 1
            GO TO 500
      200 DO 16 I = 1,18
          W 1(I) = S1(I)

```


16 $W2(I) = S2(I)$
 $W0 = S0$

C

C CALCULUL COEFICIENTILOR FOURIER COMPLEXI

C

$RC0 = U0$

$IC0 = W0$

DO 17 I = 1,18

$RCP(I) = U1(I) + W2(I)$

$ICP(I) = W1(I) - U2(I)$

$RCN(I) = U1(I) - W2(I)$

17 $ICN(I) = W1(I) + U2(I)$

C

C SCRIEREA REZULTATELOR

C

WRITE (3, 2)

WRITE (3, 3)

WRITE (3, 4)

DO 18 I = 1,86

18 WRITE (3, 5) PAC(I), X(I), PAC(I), Y(I)

WRITE (3, 6)

WRITE (3, 7) RC0, IC0

DO 19 I = 1,18

19 WRITE (3, 8) PAC(I), RCP(I), ICP(I), TAC(I),

1 RCN(I), ICN(I)

STOP

END

23. EXERCIIȚII PROPUSE

1. Să se calculeze aproximativ folosind metoda iterativă rădăcina reală a ecuației $x^3 = x + 3$. Să se folosească relația Aitken pentru accelerarea convergenței șirului de iterații.

2. Să se precizeze folosind metoda iterativă rădăcina ecuației $x = 2e^{-x/2}$ situată în vecinătatea punctului M_0 de coordonată $x_0 = 1$.

3. Să se folosească metoda lui Gauss pentru determinarea soluției sistemului de ecuații liniare

$$x_1 - x_2 + x_3 = 2$$

$$2x_1 + x_2 - x_3 = 1$$

$$x_1 + 3x_2 - x_3 = 4$$

4. Să se folosească metoda iterativă pentru precizarea soluției sistemului :

$$x = 0,7 \sin x + 0,2 \cos y$$

$$y = 0,7 \cos x - 0,2 \sin y$$

situată într-o vecinătate a punctului $M_0(0,5, 0,5)$.

5. Să se folosească metoda puterii pentru determinarea valorii proprii dominante a matricei :

$$A = \begin{pmatrix} 2 & -3 & 0 \\ -3 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$$

6. Să se calculeze polinomul de interpolare al lui Lagrange de gradul doi ce aproximează funcția dată prin tabela de valori :

x_k	0	4	9
y_k	0	2	3

Folosind polinomul calculat să se interpoleze funcția pentru $x = 1$ și să se calculeze eroarea efectuată, ținând seama că tabela de valori reprezintă în cele trei puncte funcția $y = \sqrt{x}$.

7. Să se calculeze polinomul de interpolare Newton de gradul doi cu diferențe finite la dreapta ce aproximează funcția dată prin tabela de valori :

x_k	-2	0	2
y_k	2	0	2

Folosind polinomul calculat să se extrapoleze funcția pentru $x = 3$ și să se calculeze eroarea efectuată ținând seama că tabela de valori reprezintă în cele trei puncte funcția $y = |x|$.

8. Să se calculeze dreapta de regresie ce aproximează graficul funcției dată prin tabela de valori :

x_k	-2	-1	0	1	2	3
y_k	2	1	0	1	2	3

9. Să se calculeze un polinom trigonometric de aproximare pentru funcția dată printr-o tabelă de valori într-un număr par de puncte :

x_k	0	1	2	3
y_k	0	2	2	0

Să se folosească apoi polinomul trigonometric calculat pentru a determina o valoare aproximativă a derivatei funcției în punctul $x = 2$.

10. Să se determine polinomul de minimax de gradul întâi ce aproximează funcția $f(x) = x^3$ pe intervalul $[0,4]$.

11. Să se aproximeze ecuația $F(y) = y^3$ cu un polinom de minimax de gradul întâi pe intervalul $[2, 4]$, folosind dezvoltarea funcției într-o serie formată pe baza polinoamelor Cebîșev.

12. Să se determine o soluție în sensul celor mai mici pătrate pentru sistemul supradeterminat :

$$x_1 - x_2 + x_3 = 2$$

$$2x_1 + x_2 - x_3 = 1$$

$$x_1 + 3x_2 - x_3 = 3$$

$$x_1 + 4x_2 - 2x_3 = 4$$

13. Să se determine valoarea maximă a funcției obiectiv

$$f(x_1, x_2) = 2x_1 + 5x_2$$

astfel încît să fie verificate restricțiile :

$$x_1 + x_2 \leq 10$$

$$x_1 - x_2 \geq -5$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

14. Să se calculeze derivata funcției definită prin tabela de valori

x_k	0	1	2	3	4
y_k	0	1	4	9	16

în punctul $x = 2$, folosind polinomul Newton de interpolare cu diferențe finite la dreapta. Să se calculeze eroarea efectuată, ținînd seama că tabela de valori reprezintă în cele cinci puncte date funcție $y = x^2$.

15. Folosind formula lui Simpson, să se calculeze o valoare aproximativă a integralei definite $\int_0^4 x^2 dx$, utilizînd o diviziune a intervalului $[0, 4]$ în 4 subintervale egale. Să se calculeze eroarea efectuată.

16. Folosind dezvoltarea în serie de puteri a funcției integrant să se calculeze o valoare aproximativă a integralei

$$\int_1^5 \frac{\ln(1+x)}{x} dx$$

17. Să se calculeze o valoare aproximativă a volumului mărginit de suprafața $z = 1 - x - y$ și de planele de ecuații $x = 0$, $x = 1/2$, $y = 0$, $y = 1/2$, aproximînd volumul prin integrala dublă:

$$V = \int_0^{1/2} dx \int_0^{1/2} (1 - x - y) dy$$

18. Folosind metoda Picard (metoda aproximațiilor succesive) să se determine o valoare aproximativă a soluției ecuației diferențiale $y' = x + y$, cu condiția inițială $y(0) = 2$.

19. Să se determine valoarea numerică aproximativă a soluției ecuației diferențiale $y' = x - y$ cu condiția inițială $y(0) = -1$ în punctul $x = 0,4$, folosind Metoda Euler cu pasul $h = 0,2$.

20. Să se determine valoarea numerică aproximativă a soluției ecuației diferențiale $y' = 5y$ cu condiția inițială $y(1) = 3$ în punctul $x = 1,5$, folosind metoda Runge-Kutta cu pasul $h = 0,5$.

21. Să se determine valoarea numerică aproximativă a soluției sistemului de ecuații diferențiale de ordinul întâi

$$y' = z$$

$$z' = 5z - 4y$$

în condițiile inițiale $y(0) = 0$, $z(0) = 1$, în punctul $x = 0,4$, folosind algoritmul Euler cu pasul $h = 0,2$.

22. Să se determine valoarea numerică aproximativă a soluției sistemului de ecuații diferențiale de ordinul întâi

$$y' = z$$

$$z' = 4y$$

cu condițiile inițiale $y(0) = 1$, $z(0) = 0$, în punctul $x = 0,5$, folosind algoritmul Runge-Kutta cu pasul $h = 0,5$.

23. Să se determine o soluție de forma unei serii de puteri pentru ecuația diferențială de ordinul doi $y'' = y + x$, cu condițiile inițiale $y(0) = 0$, $y'(0) = 2$.

24. Să se determine valoarea numerică aproximativă a soluției ecuației $y'' - 5y' + 4y = 0$ cu condițiile inițiale $y(0) = 0$, $y'(0) = 1$, în punctul $x = 0,5$, reducând ecuația la un sistem de ecuații de ordinul întâi și folosind pentru rezolvarea sistemului metoda lui Euler cu pasul $h = 0,5$.

25. Să se calculeze soluția ecuației diferențiale de ordinul doi $y'' = 5y$ cu condițiile la limită $y(0) = 1$, $y(1) = 0$, în punctul $x = 0,5$, folosind aproximarea lui y'' cu ajutorul diferențelor finite și împărțind intervalul $[0, 1]$ în 4 părți egale.

26. Să se determine o valoare numerică aproximativă pentru soluția ecuației lui Laplace: $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$, $0 \leq x \leq 1$, $0 \leq y \leq 1$ cu condițiile la limită $u(0, y) = u(1, y) = u(x, 0) = 0$, $u(x, 1) = 1$ prin metoda rețelei, folosind o rețea pătrată cu pasul $h = 0,25$.

27. Se dă ecuația $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$, unde funcția soluție $u(x, y)$ este definită într-un domeniu de formă triunghiulară cu vîrfurile în punctele $O(0,0)$, $A(1,0)$ și $B(0,1)$. Se dau condițiile la limită $u(0, y) = u(x, 0) = 0$ iar valoarea lui u în fiecare punct situat pe latura AB a triunghiului este egală cu 1. Să se determine o valoare numerică aproximativă pentru funcția soluției $u(x, y)$ folosind metoda rețelei și lucrînd cu o rețea pătrată avînd pasul $h = 0,25$.

28. Să se rezolve ecuația recurentă liniară de ordinul doi neomogenă :

$$y_{k+2} - 5y_{k+1} + 4y_k = k^2$$

cu condițiile inițiale : $y_0 = 0$, $y_1 = 0$.

29. Să se estimeze eroarea care se comite cînd suma seriei numerice convergente $\sum_{n=1}^{\infty} \frac{n}{(n+1)4^n}$ este aproximată prin suma primilor 10 termeni.

30. Să se folosească transformarea lui Euler cu diferențe finite pentru a mări rapiditatea convergenței seriei

$$\frac{1}{e} = 1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \dots$$

apoi să se folosească primii patru termeni ai seriei cu diferențe finite obținute pentru a obține o valoare aproximativă pentru $1/e$.

31. Folosind 4 termeni ai dezvoltării în serie MacLaurin pentru funcția $\cos x$, să se calculeze aproximativ integrala

$$\int_1^2 \frac{\cos x - 1}{x} dx$$

apoi să se evalueze eroarea comisă în acest caz.

32. Să se scrie o subrutină de calcul în limbajul FORTRAN pentru calculul numeric al soluției unei ecuații diferențiale de ordinul întâi însoțită de o condiție inițială, folosind algoritmul lui Euler.

33. Să se scrie o subrutină de calcul în limbajul FORTRAN pentru generarea unei secvențe de 10 000 de numere pseudoaleatoare.

BIBLIOGRAFIE

1. Bronson, R. *Diferential Equations*. Mc. Graw-Hill Book Company.
2. Carnahan, B. ș.a. *Applied Numerical Methodes*. John Wiley & Sons.
3. Dodescu, Gh. *Metode numerice în algebră*. București, Editura Tehnică, 1979.
4. Filipescu, D. ș.a. *Matematici generale pentru subingineri*. Culegere de probleme. București, Ed. Didactică și Pedagogică, 1975.
5. Larionescu, D. *Metode numerice și programarea în limbajul FORTRAN*. Institutul Politehnic București, 1985.
6. Bogdan, R. C. ș.a. *Sinteza mecanismelor plane articulate*. București, Ed. Academiei R.S.R., 1977.
7. Nicolae, P., Otlăcan, P. *Elemente de analiză numerică și programarea calculatoarelor electronice*. București, Academia Militară, 1971.
8. Scheid, Fr. *Numerical Analysis*. Mc. Grow-Hill Book Company.
9. Stănășilă, O. *Analiză matematică*. București, Ed. Didactică și Pedagogică, 1981.
10. Șabac, I. Gh. ș.a. *Matematici speciale*. București, Ed. Didactică și Pedagogică, 1983.
11. Salvadori, M. G., Baron, M. I. *Metode numerice în tehnică*. București, Ed. Tehnică, 1972.
12. Toma, M., Ogădescu, I. *Metode numerice și subrutine*. București, Ed. Tehnică, 1980.
13. Vraciu, G., Popa, A. *Metode numerice cu aplicații în tehnica de calcul*. Craiova, Scrisul Românesc, 1982.
14. *Mica enciclopedie matematică*. București, Ed. Tehnică, 1980.